Infotech Signer Common Criteria Evaluation

Handbuch / Guidance Document

Infotech GmbH

Autor:	Stephan Slabihoud, TÜVIT GmbH Frank Raub, Infotech GmbH Christian Weber, Infotech GmbH
Kategorie:	CC Evaluation
Status:	Final
Version:	0.5
Letzte Änderung:	2013-04-16
Dateiname:	ZEDAL_AGD_0.5.doc

Auszug

Dieses Dokument beinhaltet Benutzerhandbuch (AGD) der Komponente „Infotech Signer“.

Stichwörter

CC, AGD, Common Criteria

Revisionshistorie

Datum	Version	Autor	Bemerkung
14-Oct-08	0.1	Infotech GmbH	Erstellt
5-Jun-09	0.2	Infotech GmbH	Ergänzungen
30-Jun-09	0.3	Infotech GmbH	Interne Revision
15-Jan-10	0.4	Infotech GmbH	SSEE hinzugefügt, Formatkorrektur, Plattform hinzugefügt
22-Sep-10	0.5	Infotech GmbH	Kleine Korrektur an Multisignaturdialog, Zertifikatsanzeige, Screenshots

Diese Seite wurde absichtlich leer gelassen

Inhaltsverzeichnis

1 EINLEITUNG 9
2 INSTALLATION UND PROGRAMMSTART 10
2.1 Voraussetzungen 10
2.2 Auslieferung der SAK 11
2.3 Vorbereitungen zur Installation 12
2.4 Programminstallation 12
2.5 Überprüfung der Integrität 16
2.6 Starten der SAK 16
2.7 Wartung und Pflege 17
2.8 Uninstallation 17
3 BENUTZERHANDBUCH - ALLGEMEINES 18
3.1 Benutzerrollen 18
3.2 Betriebsmodi 18
3.3 Sichere Umgebung 18
4 BENUTZERHANDBUCH FÜR DEN ENDANWENDER 20
4.1 Sicherheitsfunktionen 20
4.1.1 SF1 - Hashberechnung und elektronische Signaturen 21
4.1.2 SF2 - Hashverifikation und elektronische Signaturen 23
4.1.3 SF3 - Eindeutige Anzeige der zu signierenden Daten 25
4.1.4 SF4 - Schutz vor Hashwert Manipulationen 25
4.1.5 SF5 - Behandlung von OCSP Daten zur Zertifikatsvalidierung 25
4.1.6 SF6 - Anbringung von Zeitstempeln 26
4.1.7 SF7 - Validierung von Zeitstempeln 27
4.1.8 SF8 - Verwaltung der SFs mit der Konfigurationsdatei 27
4.1.9 SF9 - Schutz vor Manipulation der Programmdatei 28
4.2 Benutzerschnittstelle 28
4.2.1 TrayIcon 28
4.2.2 Informationsdialog 29
4.2.3 Statusanzeige 30
4.2.4 Multisignatureinstellungen 40
4.2.5 Detailanzeige 41
4.2.6 Dateiexporte 51
4.3 Sicherheitshinweise und -empfehlungen 58
4.3.1 Fehleranzeige und -behandlung 58
4.3.2 Automatische Prüfung von Dokumenten nach der Signatur 67
4.3.3 Überprüfung von Dokumenten vor der Signatur 68
4.3.4 Überprüfung von Dokumenten nach der Signatur 68
5 BENUTZERHANDBUCH FÜR DEN ENTWICKLER 69
5.1 Sicherheitsfunktionen 69
5.2 Benutzerschnittstelle 69
5.3 Sicherheitshinweise und -empfehlungen 69

5.3.1 Fehlermeldung und -behandlung 69
5.3.2 Wahl geeigneter Metadaten 71
6 ANHANG 72
6.1 Datendefinitionen 72
6.1.1 XML.CONTENT 72
6.1.2 XML.DOCUMENT 73
6.1.3 XML.SIGNATURE 73
6.1.4 XML.CERTIFICATE 77
6.1.5 XML.ATTRIBCERT 78
6.1.6 XML.OCSP 80
6.1.7 XML.TIMESTAMP 80
6.1.8 XML.STOREPARAM 81
6.1.9 XML.PATH 84
6.1.10 DOCUMENT.FORMAT 84
6.1.11 DOCUMENT.STATUS 85
6.1.12 SIGNATURE.STATUS 85
6.1.13 SIG.TIMESOURCE 85
6.1.14 OCSP.STATUS 86
6.1.15 REVOCATION.STATUS 86
6.1.16 CERTOCSP.STATUS 86
6.1.17 CERTIFICATE.STATUS 87
6.1.18 VERIFICATION.STATUS 87
6.1.19 TIMESTAMP.STATUS 88
6.1.20 SAK.STATUS 88
6.1.21 READER.STATUS 89
6.1.22 CARD.STATUS 89
6.1.23 BINARY.EXPORT 90
6.1.24 SIGNATURE.METHOD 90
6.1.25 HASH.METHOD 91
6.2 Applikationsschnittstelle 91
6.2.1 Grundlagen 92
6.2.2 TSF-Protokoll 92
6.2.3 Basis-Kommunikationsprotokoll 92
6.2.4 Aktivierungsphase 93
6.2.5 Funktionen der SAK 94
6.2.6 Statusübergänge 102

Tabellen

Seite
Tabelle 6.1 - Einfache Datenformate ... 72
Tabelle 6.2 - DOCUMENT.FORMAT ... 84
Tabelle 6.3 - DOCUMENT.STATUS ... 85
Tabelle 6.4 - SIGNATURE.STATUS ... 85
Tabelle 6.5 - SIG.TIMESOURCE ... 85
Tabelle 6.6 - OCSP.STATUS ... 86
Tabelle 6.7 - REVOCATION.STATUS ... 86
Tabelle 6.8 - CERTOCSP.STATUS ... 86
Tabelle 6.9 - CERTIFICATE.STATUS ... 87
Tabelle 6.10 - VERIFICATION.STATUS ... 87
Tabelle 6.11 - TIMESTAMP.STATUS ... 88
Tabelle 6.12 - SAK.STATUS ... 88
Tabelle 6.13 - READER.STATUS ... 89
Tabelle 6.14 - CARD.STATUS ... 89
Tabelle 6.15 - BINARY.EXPORT ... 90
Tabelle 6.16 - SIGNATURE.METHOD ... 90
Tabelle 6.17 - HASH.METHOD ... 91
Tabelle 6.18 - Vordefinierte Selektoren ... 97
Tabelle 6.19 - Statusübergänge ... 102

Abbildungen

Seite

Abbildung 2.1 - Screenshot Setup - Einleitung 13
Abbildung 2.2 - Screenshot Setup - Installationsart 13
Abbildung 2.3 - Screenshot Setup - Verzeichniswahl 14
Abbildung 2.4 - Screenshot Setup - Shortcut 14
Abbildung 2.5 - Screenshot Setup - Installationsprotokoll 15
Abbildung 2.6 - Screenshot Setup - Fertigstellung 15
Abbildung 4.1 - Screenshot TrayIcon 29
Abbildung 4.2 - Screenshot TrayMenü 29
Abbildung 4.3 - Screenshot TrayMenü - erweitert 29
Abbildung 4.4 - Screenshot Informationsdialog 30
Abbildung 4.5 - Screenshot Statusanzeige - Initialisierung 30
Abbildung 4.6 - Screenshot Statusanzeige - Inventarisierung 31
Abbildung 4.7 - Screenshot Statusanzeige - Applikationsanmeldung 31
Abbildung 4.8 - Screenshot Statusanzeige - Applikationsgrafik 32
Abbildung 4.9 - Screenshot Statusanzeige - Datenblöcke geladen 33
Abbildung 4.10 - Screenshot Statusanzeige - Attributzertifikat laden 34
Abbildung 4.11 - Screenshot Dateiauswahldialog (Fachapplikation) 35
Abbildung 4.12 - Screenshot Statusanzeige - Attributzertifikat geladen 35
Abbildung 4.13 - Screenshot Statusanzeige - Signaturbeginn 36
Abbildung 4.15 - Screenshot Statusanzeige - Signatur ausgeführt 37
Abbildung 4.16 - Screenshot Statusanzeige - Verifikationsbeginn 38
Abbildung 4.17 - Screenshot Statusanzeige - Verifikation durchgeführt 38
Abbildung 4.18 - Screenshot Multisignatureinstellungen 40
Abbildung 4.19 - Screenshot Detailanzeige 42
Abbildung 4.20 - Screenshot Detailanzeige - Navigationsbaum 42
Abbildung 4.21 - Screenshot Detailanzeige - Dokument 44
Abbildung 4.22 - Screenshot Detailanzeige - Signatur 45
Abbildung 4.23 - Screenshot Detailanzeige - Zertifikat 47
Abbildung 4.24 - Screenshot Detailanzeige - Attributzertifikat 49
Abbildung 4.25 - Screenshot Detailanzeige - Zeitstempel 51
Abbildung 4.26 - Screenshot Export - Verzeichniswahl 52
Abbildung 4.29 - Screenshot Statusanzeige - Konfigurationsdatei veraltet 60
Abbildung 4.30 - Screenshot Statusanzeige - Parametersatz unvollständig 61
Abbildung 4.31 - Screenshot Statusanzeige - Signaturkarte nicht gelesen 61
Abbildung 4.32 - Screenshot Statusanzeige - Schlüssellänge unbestimmt 62
Abbildung 4.33 - Screenshot Statusanzeige - Sicherheitseignung abgelaufen 62
Abbildung 4.34 - Screenshot Statusanzeige - Zertifikat nicht qualifiziert 63
Abbildung 4.35 - Screenshot Statusanzeige - Signaturzertifikat nicht vertrauenswürdig 63
Abbildung 4.36 - Screenshot Statusanzeige - Signaturzertifikat noch nicht gültig 64
Abbildung 4.37 - Screenshot Statusanzeige - Signaturzertifikat nicht mehr gültig 64
Abbildung 4.38 - Screenshot Statusanzeige - Signaturzertifikat nicht lesbar 65
Abbildung 4.39 - Screenshot Statusanzeige - Attributzertifikat Ladefehler 65

Abbildung 4.40 - Screenshot Statusanzeige - Attributzertifikat Referenzfehler 66
Abbildung 4.41 - Screenshot Statusanzeige - Attributzertifikat noch nicht gültig 66
Abbildung 4.42 - Screenshot Statusanzeige - Attributzertifikat nicht mehr gültig..... 66
Abbildung 4.43 - Screenshot Statusanzeige - Signaturausführungsfehler 67
Abbildung 4.44 - Screenshot Statusanzeige - Urdokument verändert..... 67
Abbildung 6.1 - Statusübergänge 102

Diese Seite wurde absichtlich leer gelassen

1 Einleitung

Dieses Dokument ist obligatorisch für den Evaluierungs- und Zertifizierungsprozess des TOE nach den CC. Es deckt die Anforderungen bezüglich der Vertrauenskomponente AGD_OPE.1 und AGD_PRE.1 ab.

2 Installation und Programmstart

2.1 Voraussetzungen

Der Infotech Signer in der Version V2.0 / Win32 ist eine Signaturanwendungskomponente (SAK) zur signaturgesetzkonformen Erstellung und Verifikation qualifizierter Signaturen nach XML-DSig und nach RSA PKCS#7.

Die SAK wurde zur Verwendung 32-Bit Betriebssystemen von Microsoft entwickelt. Als Arbeitsumgebung wird die korrekte Installation eines der folgenden Systeme vorausgesetzt:

- Windows XP
- Windows Vista
- Windows 7

Der Rechner, auf dem die SAK betrieben werden soll, muss über einen Intel 586 kompatiblen Prozessor und sollte zum Programmstart über mindestens 128 MB freien Arbeitsspeicher (RAM) verfügen. Die Dateien der SAK belegen etwa 6 Megabyte Festplattenplatz.

Zur Anfertigung von Signaturen muss mindestens ein Kartenlesegerät und eine Signaturkarte vorhanden sein. Die zum Betrieb des Kartenlesegeräts erforderlichen Treiber müssen installiert sein.

Die zur Verwendung geeigneten Kartenlesegeräte müssen über eine Tastatur zur sicheren PIN Eingabe verfügen. Die SAK unterstützt die Verwendung folgender Kartenlesegeräte:

- Reiner SCT cyberJack e-com, Version 2.0 (TUVIT.09363.TE.06.2002)
- Reiner SCT cyberJack pinpad, Version 3.0 (TUVIT.93107.TU.11.2004)

Folgende bestätigte Signaturkarten werden unterstützt:

D-Trust, TC TrustCenter:

- Chipkarte mit Prozessor SLE66CX322P, CardOS V4.3B Re_Cert mit Applikation für digitale Signatur (T-Systems.02182.TE.11. 2006)

S-Trust:

- ZKA Banking Signature Card, Version 6.6; Giesecke & Devrient GmbH (TUVIT.93130.TU.05.2006)

Signtrust:

- STARCOS 3.2 QES Version 1.1; Giesecke & Devrient GmbH (BSI.02102.TE.11.2008)

TeleSec:

- TCOS 3.0 Signature Card, Version 1.1 (NetKey 3.0) or TCOS 3.0 Signature Card, Version 1.1 (NetKey 3.0M)
  (TUVIT.93146.TE.12.2006)

Die vorgenannten Kartenlesegeräte und Signaturkarten genügen den Anforderungen des Signaturgesetztes. Die zugehörigen Bestätigungsurkunden sind in den Veröffentlichungen der Bundesnetzagentur verfügbar (siehe http://www.bundesnetzagentur.de).

Der Betrieb mit anderen Kartenlesegeräten oder Signaturkarten ist möglich, liegt jedoch außerhalb der Bestätigung der SAK.

Die SAK gestattet während der Signaturausführung die Anbringung von Zeitstempeln an die signierten Dokumente und die Statusüberprüfung verwendeter Zertifikate beim Aussteller. Beide Funktionen bedürfen der Unterstützung der Fachapplikation. Diese leitet die von der SAK formulierten Anfragen an die jeweiligen Responder weiter und liefert die Antworten an den Infotech Signer zurück.

Dazu ist eine Online-Verbindung zu den Respondern erforderlich, die den Rechner, auf dem die SAK ausgeführt wird, mit dem Internet verbindet. Derartige Verbindungen sind über Firewalls abzusichern, die den Rechner vor unberechtigten Zugängen schützen und die Integrität des installierten Betriebssystems sicherstellen. Näheres dazu erfahren Sie im Kapitel 3.3, Sichere Umgebung.

2.2 Auslieferung der SAK

Die SAK wird in der Regel zusammen mit spezialisierten Fachanwendungen ausgeliefert. Die Fachanwendung nutzt sie wie eine Signaturbibliothek.

Die Auslieferung erfolgt i.d.R. zusammen mit der Fachanwendung per Webdownload von der Firmenhomepage des Unternehmens. Der Download findet über einen SSL geschützten Kanal statt. Über das Serverzertifikat kann der Anwender die Identität des Servers kontrollieren.

Die SAK ist auch als Entwicklerversion zur Verwendung in weiteren Fachanwendungen verfügbar. Die Entwicklerversion wird auf Anfrage mit weiteren Dokumentationen und Klassen zur Integration in Anwendungen auf einem Installationsmedium (CD-ROM) ausgeliefert.

Bei Bezug über ein Installationsmedium ist sicherzustellen, dass die Sicherheitsmerkmale der Verpackung (Laminierungen und Siegel) unbeschädigt sind. Wenn Beschädigungen erkennbar sind, darf das Installationsmedium nicht verwendet werden. Sobald das Medium der Verpackung entnommen ist, muss es vor dem Zugriff unberechtigter Dritter geschützt werden.

2.3 Vorbereitungen zur Installation

Benutzer sowie ggf. Administratoren und Wartungspersonal mit Zugang zu den Installationsdateien oder den installierten Dateien müssen vertrauenswürdig sein und den Anweisungen der Benutzerdokumentation Folge leisten.

Vor der Installation ist sicherzustellen, dass das Betriebssystem frei von jedweder Schadsoftware (Viren, Trojaner und Backdoor-Programme) ist. Dazu sind geeignete Virenscanner mit aktuellen Virendefinitionsdateien zu verwenden.

Wenn die Installation aus einem Backup wiederhergestellt wurde, ist ihre Integrität in jedem Fall vor der ersten Inbetriebnahme sicherzustellen (siehe 2.5, Überprüfung der Integrität).

Des Weiteren sind für den gesamten Installationsprozess die im Abschnitt 3.3, Sichere Umgebung, genannten Maßnahmen einzuhalten.

2.4 Programminstallation

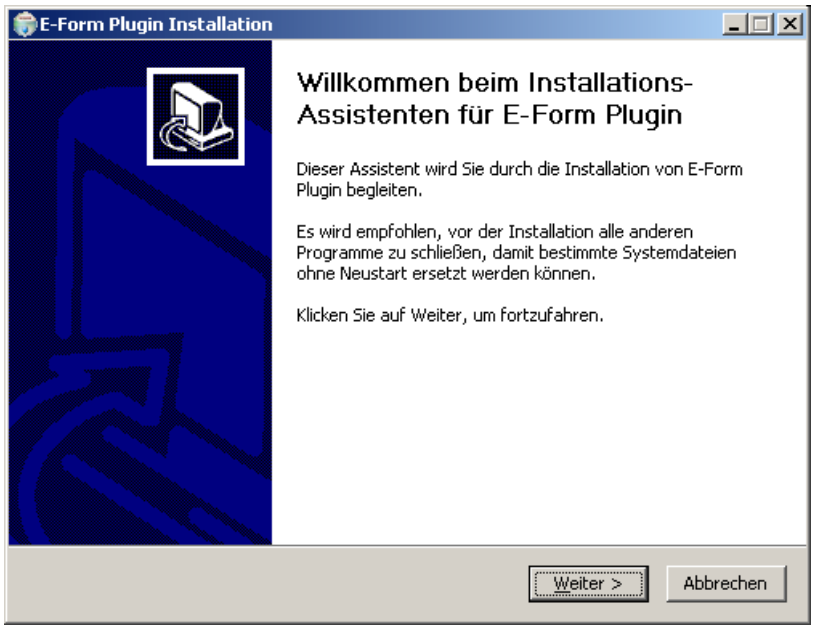
Bei der Installation kann, je nach Installationsstand des Betriebssystems, die Komplettierung der erforderlichen Laufzeitbibliotheken notwendig sein. Diese wird durch das Setupprogramm im Rahmen der Installation vorgenommen.

Die Komponenten der SAK werden in ein Verzeichnis, i.d.R. zusammen mit der Fachanwendung, installiert. Das Verzeichnis kann im Rahmen des Setupprogramms frei gewählt werden.

Die Installation beginnt mit der Aktivierung des Installationsprogramms. Dieses trägt i.d.R. den Namen Setup.exe. Es gibt eine Vielzahl möglicher Setupprogramme. Darunter auch solche, die für automatische Installationen (und Updates) vorgesehen sind (“unattended“ oder “silent“ Setups). Auch bei solchen Setupprogrammen ist die Integrität der SAK vor ihrer ersten Verwendung durch geeignete Maßnahmen sicherzustellen.

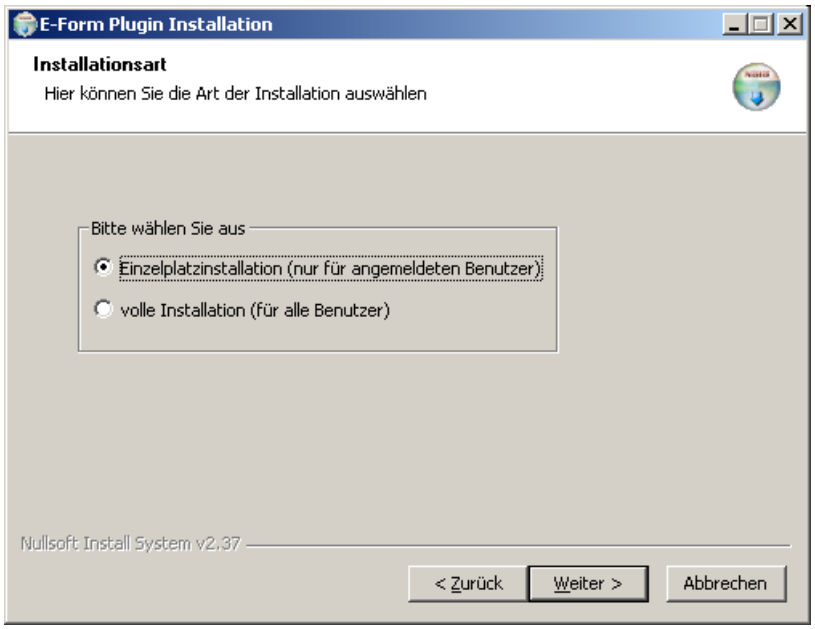
Im Folgenden wird der Installationsablauf am Beispiel einer interaktiven Installation der Fachanwendung E-Form Plugin dargestellt. Das Setupprogramm zeigt zunächst eine kurze Einleitung:

Abbildung 2.1 - Screenshot Setup - Einleitung


E-Form Plugin Installation
Willkommen beim Installations-Assistenten für E-Form Plugin
Dieser Assistent wird Sie durch die Installation von E-Form Plugin begleiten.
Es wird empfohlen, vor der Installation alle anderen Programme zu schließen, damit bestimmte Systemdateien ohne Neustart ersetzt werden können.
Klicken Sie auf Weiter, um fortzufahren.
Weiter >
Abbrechen

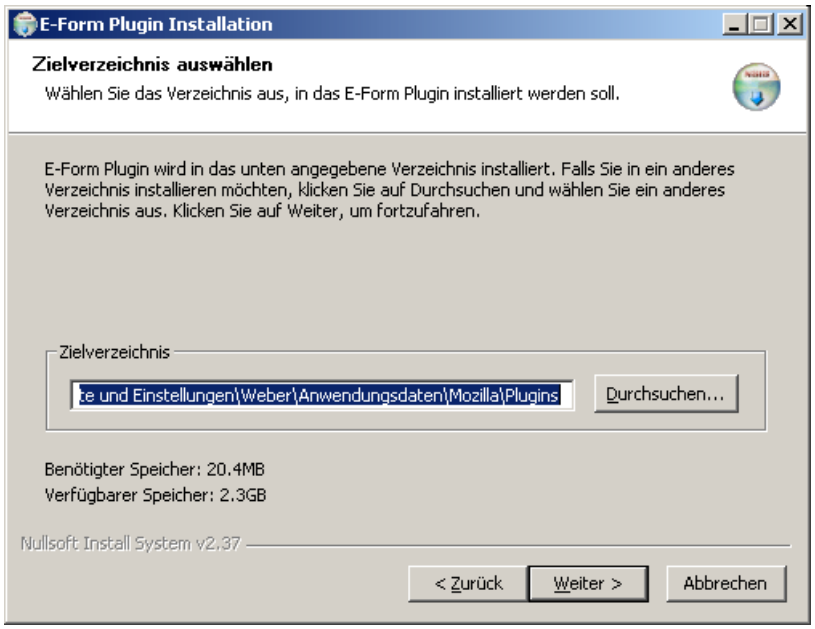
Die Installation kann für einen einzelnen Benutzer oder für alle Benutzer der Arbeitsstation vorgenommen werden. Der folgende Dialog bietet die entsprechenden Auswahlmöglichkeiten:

Abbildung 2.2 - Screenshot Setup - Installationsart



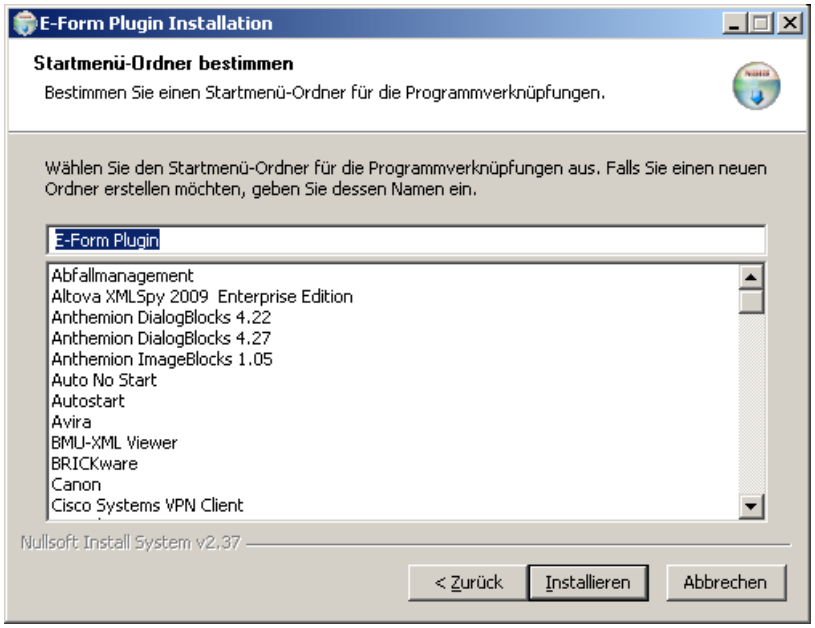
Die Wahl des Zielverzeichnisses richtet sich nach den Anforderungen der Fachapplikation, kann aber zur Anpassung an spezielle Gegebenheiten angepasst werden:

Abbildung 2.3 - Screenshot Setup - Verzeichniswahl


E-Form Plugin Installation
Zielverzeichnis auswählen
Wählen Sie das Verzeichnis aus, in das E-Form Plugin installiert werden soll.
E-Form Plugin wird in das unten angegebene Verzeichnis installiert. Falls Sie in ein anderes Verzeichnis installieren möchten, klicken Sie auf Durchsuchen und wählen Sie ein anderes Verzeichnis aus. Klicken Sie auf Weiter, um fortzufahren.
Zielverzeichnis
te und Einstellungen\Weber\Anwendungsdaten\Mozilla\Plugins
Durchsuchen...
Benötigter Speicher: 20.4MB
Verfügbarer Speicher: 2.3GB
Nullsoft Install System v2.37
< Zurück
Weiter >
Abbrechen

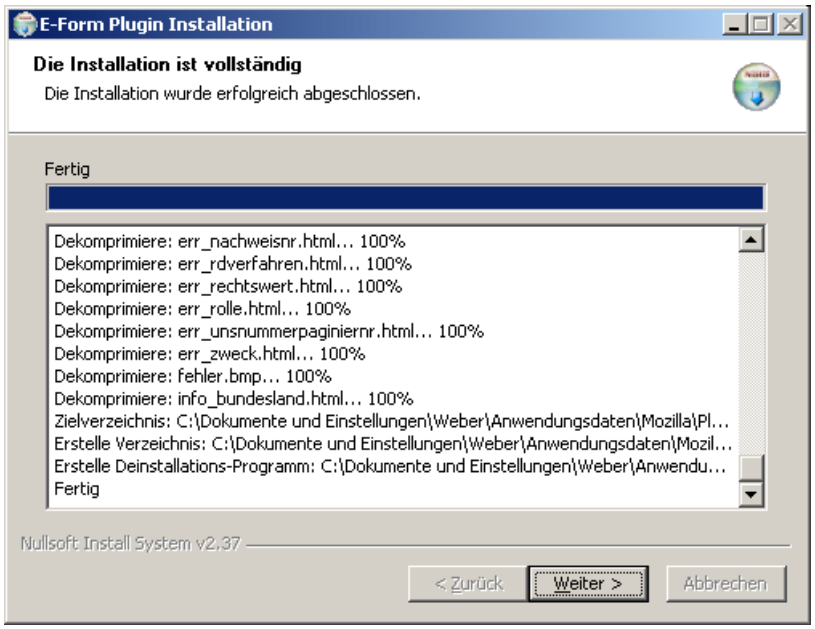
Die meisten Setupprogramme bieten die Möglichkeit Shortcuts zum Programmstart anzulegen oder den Programmstart über das Startmenü verfügbar zu machen:

Abbildung 2.4 - Screenshot Setup - Shortcut



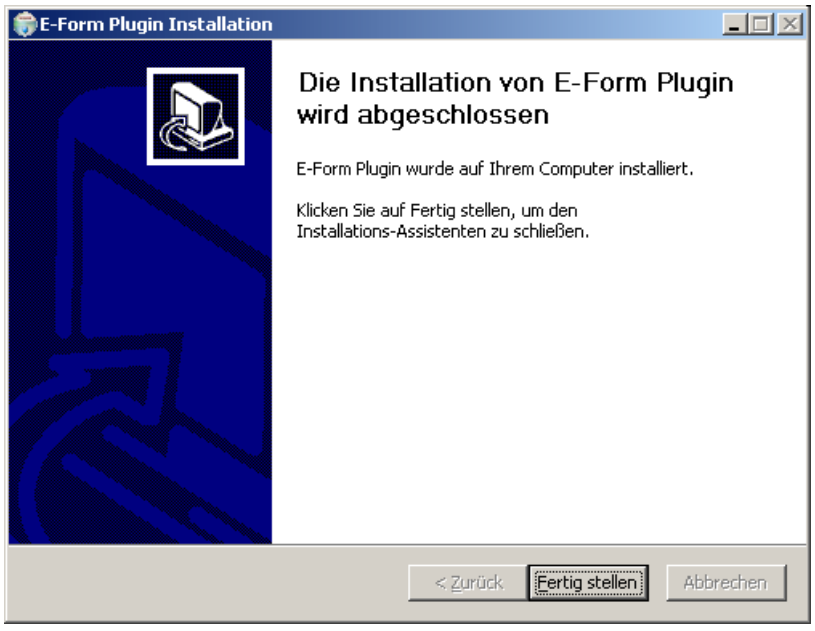
Nach Aufnahme der änderbaren Installationsparameter wird die Installation durchgeführt und der Installationsfortschritt üblicherweise angezeigt und ggf. ein Installationsprotokoll angefertigt.

Abbildung 2.5 - Screenshot Setup - Installationsprotokoll



Zum Abschluss der Installation wird eine Meldung angezeigt, die über den Erfolg des Installationsvorgangs informiert. Die Bestätigung der Abschlussmeldung beendet das Setupprogramm.

Abbildung 2.6 - Screenshot Setup - Fertigstellung


E-Form Plugin Installation
Die Installation von E-Form Plugin wird abgeschlossen
E-Form Plugin wurde auf Ihrem Computer installiert.
Klicken Sie auf Fertig stellen, um den Installations-Assistenten zu schließen.
< Zurück
Fertig stellen
Abbrechen

Nach erfolgter Installation ist der Zugriff und die Verwendung der SAK durch den Benutzer sicherzustellen. Der Computer-Administrator räumt dem Endanwender die dazu nötigen Rechte ein.

2.5 Überprüfung der Integrität

Die SAK besteht aus einer ausführbaren Programmdatei, einer Treiber-dll für Kartenlesegeräte und einer Konfigurationsdatei. Zur Prüfung der Integrität der Installation ist jede dieser drei Dateien mit einer separaten Signaturdatei signiert. Jede Signaturdatei trägt denselben Namen wie die signierte Datei, jedoch mit der zusätzlichen Endung ".p7s". Alle 6 Dateien liegen zusammen in einem Installationsverzeichnis, das i.d.R. mit dem der Fachapplikation identisch ist.

Die Integrität der Installation muss vor der ersten Verwendung verifiziert werden. Dazu genügt die Nachberechnung des SHA-256 Hashwerts über die Programmdatei "ITSigner.exe" und Vergleich des Berechnungsergebnisses mit dem hier veröffentlichten Wert.

Der SHA-256 Hashwert der Programmdatei "ITSigner.exe" lautet:

b3ec96606cde63e161f3e8d664a38f9360b66072fb457aa96d82771b3b07baed

Das Infotech Signer Integrity Tool "ITSigner.Check.exe" berechnet die Hashwerte der signierten Dateien, prüft die Signaturen und weist die Hashwerte zusammen mit den Prüfergebnissen aus.

Der Hashwert (SHA-256) kann auch mit einer anderen geeigneten Anwendung berechnet und ebenso telefonisch erfragt werden. Telefonische Unterstützung und Auskunft über den korrekten Hashwert erhalten Sie bei der

Infotech Zentrale unter: +(49) 2361 9130 0

2.6 Starten der SAK

Die SAK wird durch die sie verwendende Fachapplikation gestartet. Der Start erfolgt je nach Applikation ggf. verzögert, spätestens jedoch bis zur ersten Ausführung oder Verifikation einer Signatur.

Sobald die SAK gestartet ist, zeigt sie in der Taskbar das Trayicon , über dessen Kontextmenü während der gesamten Programmlaufzeit die Statusanzeige und der Informationsdialog angefordert werden kann.

Unmittelbar nach Programmstart führt die SAK eine Reihe interner Prüfungen aus und zeigt deren Fortschritt auf de Statusanzeige. Näheres dazu ist im Abschnitt 4.2, Benutzerschnittstelle, zu finden.

Wenn die SAK die Signatur der beiliegenden Signaturdatei nicht verifizieren kann oder die Konfigurationsdatei veraltet ist, zeigt er diesen Sachverhalt in der Statusanzeige an und stellt jedwede Funktionalität ein. Die Anzeige wird auf Bestätigung des Benutzers oder nach längstens 15 Sekunden automatisch beendet. Siehe dazu Abschnitt 4.3, Sicherheitshinweise und -empfehlungen.

Nach erfolgreichem Abschluss der Prüfungen wird die Statusanzeige verborgen und erst auf Benutzeranforderung sowie bei Signatur- oder Verifikationsaktivität wieder angezeigt.

2.7 Wartung und Pflege

Die Integrität der Installation der SAK muss regelmäßig, mindestens jedoch monatlich, geprüft werden. Das Verfahren dazu ist im Abschnitt 2.5, Überprüfung der Integrität, beschrieben.

Die in der Konfigurationsdatei enthaltenen Informationen (Sicherheitseignungen und Zertifikatsstämme) müssen stets auf aktuellem Stand gehalten werden. Um dies gewährleisten zu können, ist die Konfigurationsdatei mit einem Ablaufdatum versehen. Wenn dieses Datum überschritten ist, weist die SAK auf diesen Sachverhalt hin und verweigert die weitere Ausführung (siehe 4.3.1.1, Startfehler).

Die Aktualisierung der Konfigurationsdatei kann von den die SAK verwendenden Applikationen unterstützt werden. In diesem Fall sorgt sie für den Download und die Hinterlegung im Dateisystem. Andernfalls muss der Benutzer die Konfigurationsdatei und die zugehörige Signaturdatei manuell aktualisieren.

Vor der Aktualisierung sind alle laufenden Instanzen der SAK zu beenden. Zudem müssen die Berechtigungen des Anwenders zur Aktualisierung hinreichen. Aktuelle Sätze aus Konfigurationsdatei und zugehöriger Signaturdatei können über die Firmenhomepage des Herstellerunternehmens (über einen sicheren Kanal) geladen werden.

2.8 Uninstallation

Die Uninstallation der SAK erfolgt i.d.R. zusammen mit der Uninstallation der Fachanwendung durch deren Setupprogramm (bzw. dem mitinstallierten Uninstaller). Sie kann aber auch manuell durch Löschung der Dateien im Installationsverzeichnis vorgenommen werden.

3 Benutzerhandbuch - Allgemeines

3.1 Benutzerrollen

Dieses Handbuch wendet sich an den Endanwender der SAK ebenso wie an den Entwickler, der die SAK in einer Fachapplikation einsetzen möchte.

Für den Endanwender enthält es die erforderlichen Informationen, um die SAK innerhalb ihrer bestätigten Umgebung sicher verwenden zu können. Es spezifiziert die erforderlichen Maßnahmen zur Herstellung der zulässigen Einsatzumgebung und den Umgang mit den Hilfsmitteln zur rechtssicheren Erstellung und Prüfung qualifizierter Signaturen.

Dem Entwickler dient es als Leitfaden, wie die Dokumentation zu seiner Fachapplikation zu ergänzen ist, um den Endanwender in die Lage zu versetzen eine sichere Einsatzumgebung herzustellen, die Integrität der SAK vertrauenswürdig zu prüfen und sie sicher zu verwenden.

3.2 Betriebsmodi

Die SAK genügt in ihrer Standardkonfiguration den Anforderungen des Signaturgesetzes und der Signaturverordnung (SigG i.V.m. SigV). Die Standardkonfiguration wird nach jedem Programmstart angenommen.

Die SAK kommuniziert zu jeder Zeit mit höchstens einer sie verwendenden Fachapplikation. Der gleichzeitige Betrieb mehrerer SAK Instanzen ist allerdings möglich. Jede die SAK verwendende Fachapplikation ist dann einer dezidierten Instanz der SAK zugeordnet.

3.3 Sichere Umgebung

Die SAK verfügt über Mechanismen zur Prüfung der Integrität des eigenen Programmcodes, die jedwede Manipulation zuverlässig feststellten. Die SAK terminiert mit einer Fehlermeldung, falls der Programmcode verändert wurde.

Um sicherzustellen, dass von der Fachapplikation das richtige Programm verwendet wird, ist die Integrität der Installation vor der ersten Inbetriebnahme und dann regelmäßig, wenigstens jedoch monatlich zu prüfen. Die SAK weist den Namen und Speicherort der eigenen Programmdatei in dem Informationsdialog aus. Es ist sicherzustellen, dass die Fachapplikation diese Programmdatei verwendet und dass eine Nachberechnung des Hashwertes über die Programmdatei den korrekten, öffentlich bekannten Wert aufweist.

Das Verfahren zur Feststellung der Integrität ist in Abschnitt 2.5, Überprüfung der Integrität, beschrieben.

Die Arbeitsstation, d.h. der Rechner und das Betriebssystem mit dem die SAK verwendet wird, muss frei von jedweder Schadsoftware (Viren, Trojaner, Backdoor-Programme) sein. Die SAK verfügt jedoch über keinerlei Mechanismus die Unversehrtheit des Betriebssystems festzustellen. Zur Diagnose sind geeignete Virenscanner mit aktuellen Virendefinitionsdateien zu verwenden. Rechner und Betriebssystem sind vor Eingriffen unberechtigter Dritter zu schützen, jede Kommunikation mit dem Internet ist über geeignet konfigurierte Firewalls zu führen.

Der Endanwender muss zu jedem Zeitpunkt die volle Kontrolle über die eingelegten Speichermedien und Netzwerkfreigaben haben. Der Installationsort der SAK darf zu keiner Zeit durch unberechtigte Dritte erreichbar sein.

Das bei Signaturvorgängen eingesetzte Kartenlesegerät ist vor Manipulationen Dritter zu schützen. Vor der Inbetriebnahme ist zu prüfen, ob die angebrachten Siegel unbeschädigt sind. Falls eine Beschädigung erkennbar ist, darf das betroffene Kartenlesegerät nicht zu Signaturzwecken verwendet werden.

Die Geheimhaltung und Unversehrtheit des privaten Schlüsselmaterials obliegt der eingesetzten Signaturkarte. Die Verwendung der Signaturkarte ist personengebunden, die PIN zur Authentisierung ist vertraulich. Zu keiner Zeit darf die PIN weitergegeben, niedergeschrieben oder sonst irgendwie bekannt gemacht werden. Wenn Zweifel an der Vertraulichkeit der PIN bestehen, ist die Signaturkarte unverzüglich beim Aussteller zu sperren.

Die Signaturkarte ist vor dem Zugriff unberechtigter Dritter zu schützen und muss sich stets im unmittelbaren Einflussbereich des Endanwenders befinden. Dieser Schutzbedarf besteht in gesteigertem Maß für Multisignaturkarten. Zur Signaturausführung freigeschaltete Signaturkarten dürfen den unmittelbaren Einflussbereich des Endanwenders zu keiner Zeit verlassen.

Die SAK trägt in den angefertigten Signaturen die Uhrzeit des Rechners ein, auf dem sie ausgeführt wird. Generell wird als Zeitbasis Datum und Uhrzeit in UTC verwendet. Die SAK verfügt über keinerlei Mechanismus zur Feststellung der korrekten Uhrzeit. Der Anwender hat die zutreffende Einstellung des Datums, der Uhrzeit und der Zeitzone auf der verwendeten Arbeitsstation sicherzustellen.

4 Benutzerhandbuch für den Endanwender

Dieses Handbuch beschreibt die Eigenschaften der SAK und enthält Screenshots typischer Anzeigen während der Programmausführung. Es enthält die zum sicheren Umgang mit der SAK erforderlichen Informationen. Im Abschnitt 4.2.6, Dateiexporte, werden tiefere Detailinformationen zu den Ausgaben vermittelt und dabei Bezug auf die im Anhang beschriebenen Definitionen genommen. Der Anhang 6.1, Datendefinitionen, ist Bestandteil des Handbuchs.

4.1 Sicherheitsfunktionen

In diesem Kapitel sind die Sicherheitsfunktionen der SAK beschrieben. Die Beschreibung ist in dem ST, Kapitel 7.1 ff., “TOE summary specification“ entsprechende Abschnitte unterteilt

- SF1 - Hashberechnung und elektronische Signaturen
- SF2 - Hashverifikation und elektronische Signaturen
- SF3 - Eindeutige Anzeige der zu signierenden Daten
- SF4 - Schutz vor Hashwert Manipulationen
- SF5 - Behandlung von OCSP Daten zur Zertifikatsvalidierung
- SF6 - Anbringung von Zeitstempeln
- SF7 - Validierung von Zeitstempeln
- SF8 - Verwaltung der SFs mit der Konfigurationsdatei
- SF9 - Schutz vor Manipulation der Programmdatei

Hinweis:

In dieser Dokumentation wird unterschieden zwischen Verifikation (der mathematischen Korrektheit) und der Validierung (Einhaltung von Profilen bei prüfbedürftigen Signaturen). Die Validierung hinsichtlich der Gültigkeitszeiträume der verwendeten Zertifikate ist bei Anwendung des Schalenmodells stets strenger als beim Kettenmodell. OCSP und TS liefern Online-Informationen und müssen der Validierung durch das Schalenmodell standhalten (mit dem Ausstellungszeitpunkt der Informationen als Prüfzeitpunkt).

Prüfungen nach dem zutreffenden Prüfmodell werden ausschließlich für Dokumentsignaturen vorgenommen, da diese sonst nach Ende des Gültigkeitszeitraums des Zertifikats ebenfalls ungültig würden (zumindest dann, wenn sie keine Zeitinformation tragen; bei bekannter Signaturausführungszeit sollten die Signaturen jedoch beiden Modellen standhalten können).

Auch historische Signaturen sollen prüfbar sein. Diese können durchaus nach ungültig gewordenen Verfahren angefertigt worden sein (und wurden ggf. übersigniert). Die Anwendung eines nicht mehr sicherheitsgeeigneten Verfahrens oder abgelaufener Zertifikate wird nicht unterbunden, jedoch dem Benutzer zur Kenntnis gebracht.

4.1.1 SF1 - Hashberechnung und elektronische Signaturen

Eine elektronische Signatur wird erstellt, indem über die zu signierenden Daten ein Hashwert berechnet und dieser Hashwert mit einem privaten Schlüssel verschlüsselt wird. Bei Signaturen nach PKCS#7v1.5 besteht die Alternative, den Hashwert über den gesamten Datenblock zu verschlüsseln (Signatur 1. Ordnung) oder die Hashmethode und den Hashwert in die authentisierten Attribute einzutragen und diese gemeinsam mit weiteren authentisierten Attributen (z.B. Signaturzeit) zu signieren (Signatur 2. Ordnung). Die SAK verwendet bei der Signaturerzeugung nach PKCS#7 stets die Signatur 2. Ordnung.

Auch XML-Signature verwendet die Signatur 2. Ordnung. Es werden zunächst die Reference-Knoten innerhalb des SignedInfo-Knotens ausgewertet, daraus Hashwerte gebildet und diese in die Reference-Knoten eingetragen. Aus dem SignedInfo-Knoten wird dann der Hashwert für die Signatur gebildet.

Zum Einsatz kommen hierbei SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 oder RIPE-MD 160. Nachdem der Hashwert berechnet wurde, initiiert die SAK dessen Signierung[1] mit Hilfe einer Signaturkarte und geeignetem Kartenleser[2] über die Schnittstelle zum Kartenleser[3].

Die elektronische Signatur wird in der Signaturkarte mit Hilfe des RSA-Algorithmus berechnet. Die SAK integriert den erhaltenen Signaturwert in den vorhandenen Datenbestand nach:

- PKCS#7
- XML Signature
- XAdES

Durch die elektronische Signatur wird die Authentizität des Datenblocks sichergestellt. Durch das zusätzliche Hinzufügen des Zertifikats ist es zudem möglich den Datenblock zu verifizieren.

Bevor die Berechnung eines Hashwerts startet, wird eine eindeutige Meldung angezeigt, dass nun eine elektronische Signatur erzeugt wird. Es wird zudem angezeigt, welche Daten signiert werden. Durch die Eingabe einer PIN am Kartenleser und dem notwendigen Besitz einer Signaturkarte ist es gewährleistet, dass nur autorisierte Personen eine elektronische Signatur erstellen können.

Die SAK erlaubt es dem signierten Datenblock optional einen Zeitstempel hinzuzufügen.

Es wird eine Massensignatur unterstützt, sofern geeignete Signaturkarten zum Einsatz kommen (siehe Folgeabschnitt).

Wenn der Anwender einen Signaturvorgang abbricht, werden alle bereits erzeugten Signaturen verworfen und alle geladenen Dokumente in den Zustand vor der Signatur versetzt.

Diese Sicherheitsfunktion benötigt eine Konfigurationsdatei (siehe SF8) mit den derzeit gültigen Algorithmen und Parametern, sowie eine Signaturkarte. Anderenfalls wird den Benutzer eine eindeutige Fehlermeldung angezeigt, dass eine Signatur nicht erstellt werden kann.

---

[1] Signierung des Hashwerts mit dem privaten Schlüssel der Karte (Signaturschlüssel).

[2] Es werden mehrere Kartenleser gleichzeitig unterstützt.

[3] Ein externer Treiber stellt hierbei die Verbindung zu den Schnittstellen PC/SC oder CT-API oder einem herstellerspezifischen Treiber her.

Die benutzte kryptographische Methode ist gemäß §17 Paragraph 1 bis 3 SigG 22.05.2001 in Verbindung mit Anhang 1, Paragraph 1 Nr. 2 SigV 22.11.2001 als geeignet identifiziert.

4.1.1.1 Verwendung von Multisignaturkarten

Neben den Standard-Signaturkarten, die für jede Signatur eine PIN-Eingabe erfordern, gibt es Multisignaturkarten, die nach der PIN-Eingabe eine beliebige Anzahl von Signaturen zulassen.

Diese Karten werden u.a. in folgenden typischen Fällen eingesetzt:

- ein Benutzer muss eine Vielzahl an Dokumenten in kurzer Zeit signieren. Die Eingabe einer PIN pro Dokument wäre mühsam, zeitaufwendig und fehlerträchtig.
- an einer Arbeitsstation ist durch Kundenverkehr in kurzen Abständen eine Signatur zu leisten. Eine PIN-Eingabe in einer solchen Umgebung würde den Arbeitsablauf stören und hätte ggf. auch die Möglichkeit des leichteren Ausspähens der PIN zur Folge.

Um diese und weitere Fälle zu unterstützen, kann die Verwendung von Multisignaturen individuell eingestellt werden. Die Einstellungen werden an das Signaturzertifikat der Signaturkarte gebunden und können nur dann verändert werden, wenn eine Signaturkarte über ein Kartenlesegerät verbunden ist.

Folgende paarweise exklusiven Einstellungen sind vorgesehen:

- Multisignaturen sperren (Standardeinstellung)
- Multisignaturen für die geladenen Dokumente zulassen
- Multisignaturen über die gesamte Programmlaufzeit zulassen
- Multisignaturen speziell zulassen, beschränkt auf:
  - maximale Zeit der Nichtnutzung zwischen Signaturausführungen, Vorgabe: 5 Minuten
  - maximale Anzahl der Gesamtsignaturen, Vorgabe: Anzahl der geladenen Dokumente

Die manuellen Einstellungen werden bei der ersten folgenden Signatur mit der Kartennummer wirksam. Bei dieser Signatur muss der Benutzer die Einstellungen bestätigen. Nach der Bestätigung der Einstellungen wird die Multisignatur aktiviert und beim ersten auftretenden Abbruchkriterium beendet. Lehnt der Benutzer die Multisignatur ab, wird eine einzelne Signatur getätigt. Die Einstellungen bleiben erhalten.

Wenn für eine Karte keine Einstellungen vorliegen, es sich um eine Multisignaturkarte handelt und mit der Karte innerhalb der Vorgabezeit (5 Minuten) bereits eine Signatur getätigt wurde, wird der Benutzer auf die Möglichkeit der Multisignatur hingewiesen. Bestätigt der Benutzer die Multisignatur, wird diese mit dem Vorgabewert von 5 Minuten Nichtbenutzung aktiviert. Bei Ablehnung wird die Multisignatur gesperrt und muss später manuell aktiviert werden.

Ist eine Ablaufzeit oder die Anzahl der Gesamtsignaturen angegeben, wird die Multisignatur nach einem dieser Abbruchkriterien gesperrt und muss manuell aktiviert werden. Wurde nur die Zeit für die Nichtbenutzung manuell eingestellt, verbleibt die Einstellung auch nach der Dauer

der Nichtbenutzung und es wird bei der nächsten Signatur sofort die Bestätigung zur Multisignatur eingeholt.

Die Multisignatur ist in jedem Fall beendet, wenn die Multisignaturkarte gezogen und damit die Verbindung zur Karte unterbrochen wurde.

4.1.2 SF2 - Hashverifikation und elektronische Signaturen

Die SAK kann elektronische Signaturen verifizieren, die auf SHA-1, SHA-224, SHA-256, SHA384. SHA-512 oder RIPE-MD 160 Hashwerten beruhen. Die Verifikation wird durchgeführt, indem der Hashwert über die zu prüfenden Daten berechnet wird, der Original-Hash aus der Signatur mit Hilfe des RSA-Algorithmus extrahiert wird und hierfür der öffentliche Schlüssel (Verifikationsschlüssel) aus dem gegebenen Zertifikat des Erstellers der Signatur zur Verifikation verwendet wird.

Hierbei wird das Zertifikat des Erstellers stets mitgeführt (in PKCS#7 und in XML Signature mit XAdES). Die SAK bekommt das zu prüfende „Dokument“ als Datenblock, sowie die Klasse des Blocks übergeben (PKCS#7 oder XML Signature mit XAdES).

PKCS#7

Im Falle von PKCS#7 wird zunächst die ggf. vorhandene base64-Transportkodierung entfernt, so dass die Rohdaten vorliegen. Auf diese wird ASN.1 angewandt und der Signaturwert extrahiert. Anschließend wird über den Datenblock der gewünschte Hashwert gebildet. Wenn die Signatur authentisierte Attribute enthält, wird geprüft, ob diese den ermittelten Hashwert enthalten und über alle authentisierten Attribute erneut der Hashwert gebildet.

Der Signaturwert wird mit dem öffentlichen Schlüssel des enthaltenen Zertifikats nach RSA entschlüsselt und der so erhaltene Hashwert wird mit dem oben berechneten verglichen.

Die Signaturzeit wird den authentisierten Attributen entnommen.

Dieser Vorgang wird wiederholt, wenn die Rohdaten wiederum im PKCS#7-Format vorliegen.

XML

Bei XML werden zunächst die Knoten ermittelt, die Signaturdaten nach dem XML-Signature-Standard enthalten. Die innerhalb der SignedInfo-Struktur enthaltenen Reference-Knoten werden ausgewertet und die XML-Daten entsprechend des XML-Signature-Standards gefiltert und nach Canonical-XML transformiert. Über die resultierenden Datenblöcke werden Hashwerte gebildet und mit den Reference-Einträgen verglichen. Die SignedInfo-Struktur wird ebenfalls nach Canonical-XML transformiert. Aus diesem wird der resultierende Hashwert berechnet. Die Signatur wird mit dem öffentlichen Schlüssel des enthaltenen Signaturzertifikats nach RSA entschlüsselt und der so erhaltene Hashwert mit dem resultierenden Hashwert verglichen.

Die Signaturzeit wird ggf. aus den erweiterten Signaturdaten nach XAdES ermittelt.

In beiden Fällen wird zusätzlich die Zertifikatskette des verwendeten Signaturzertifikats aus dem in der Konfigurationsdatei enthaltenen Zertifikatsstamm ermittelt und nach ISIS MTT oder RFC 3280 geprüft.

Die SAK zeigt eine eindeutige Meldung an, ob die berechneten Hashwerte mit den im Dokument hinterlegten identisch sind, die Signatur somit korrekt ist und die signierten Daten authentisch sind. Durch die Verwendung einer internen Anzeige ist sichergestellt, dass der Status der Signaturprüfung und die Meta-Informationen über die Daten eindeutig angezeigt werden.

Die Daten können optional für den Anwender exportiert werden, so dass sie in einem geeigneten externen Viewer zur Kontrolle nochmals angezeigt werden können. Im Falle von XML wird zu jedem gebildeten Hashwert eine eigene Datei exportiert. Das Ergebnis der Signaturprüfung wird ebenfalls der Anwendung zur Verfügung gestellt.

Die SAK bietet die Möglichkeit den Anwender, der die elektronische Signatur erstellt hat, zu identifizieren. Der Benutzer, der die elektronische Signatur mit der SAK verifiziert, kann sich Kenndaten des für die Erzeugung der Signatur verwendeten Zertifikats anzeigen lassen und es ebenfalls exportieren.

Während der Verifikation wird der Aussteller des Zertifikats ermittelt und die Gültigkeit seines Zertifikats validiert. Wenn das Zertifikat ungültig ist, wird diese Information angezeigt.

Der Benutzer kann die Einholung von Gültigkeitsinformationen zu Signaturzertifikaten bekannter Aussteller mittels OCSP veranlassen. Erhaltene OCSP-Informationen werden angezeigt und können exportiert werden.

Wenn in den Daten ein Signaturzeitstempel enthalten ist, wird dieser ebenfalls angezeigt und dazu verwendet, zu prüfen, ob das Signaturzertifikat zum Zeitpunkt der Signaturerstellung gültig war.

Grundsätzlich wird die Prüfung mit der in den Daten kodierten Signaturzeit durchgeführt. Diese ist i.d.R. die Systemzeit, zu der die Signatur ausgestellt wurde. Falls keine Zeit verfügbar ist, wird die aktuelle Systemzeit verwendet.

Hinweis:

Signaturzertifikate von Endbenutzern werden grundsätzlich nach dem Kettenmodell validiert. Bei der Validierung von Zeitstempel- oder OCSP-Responsen oder anderer Zertifikate, die nicht zur Signaturausführung gekennzeichnet sind und dem TOE zur Validierung vorgelegt werden, wird stets das Schalenmodell verwendet.

Diese Sicherheitsfunktion benötigt eine Konfigurationsdatei (siehe SF8) mit vertrauenswürdigen Zertifikatsstämmen und den derzeit gültigen Algorithmen und Parametern.

Die benutzte kryptographische Methode ist gemäß §17 Paragraph 1 bis 3 SigG 22.05.2001 in Verbindung mit Anhang 1, Paragraph 1 Nr. 2 SigV 22.11.2001 als geeignet identifiziert.

4.1.3 SF3 - Eindeutige Anzeige der zu signierenden Daten

Die SAK stellt sicher, dass der Inhalt eines Datenblocks eindeutig durch die Anzeige von Meta-Daten identifizierbar ist. Die Metainformationen beinhalten keine versteckten oder aktiven Inhalte. Prinzipiell können alle Dateitypen signiert werden, da die Signaturkomponente intern diese (bis auf XML-Daten) nur als „Datenblock“ behandelt.

Datenblock und Meta-Daten werden der SAK zusammen übergeben und in demselben Ablagefach gespeichert, so dass ein unmittelbarer Zusammenhang hergestellt werden kann und erhalten bleibt.

Bei Bedarf liegen die Daten an einer externen Schnittstelle an, die dann mit einem für dieses Format speziellen Viewer (nicht innerhalb der Evaluierung) betrachtet werden können.

Im Falle von XML werden die Daten nach Anwendung der Filter- und Transformationsregeln nach XML-Signature exportiert, die auch bei der Signatur zum Einsatz kommen. Für den SignedInfo-Block und jeden Reference-Knoten wird eine eigene Datei erstellt.

4.1.4 SF4 - Schutz vor Hashwert Manipulationen

Nachdem die elektronische Signatur erstellt wurde (siehe SF1), überprüft die SAK diese mit Hilfe des öffentlichen Schlüssels aus dem Zertifikat der Signaturkarte[4]. Wenn der Original-Hashwert und der dekodierte Hashwert aus der elektronischen Signatur nicht übereinstimmen, wurde der Hashwert bei der Übertragung von der Smartcard verändert und es wird eine eindeutige Meldung ausgegeben.

Durch den Vergleich des Hashwerts, der für die Erstellung der Signatur berechnet wurde, und dem Hashwert, der von der Signaturkarte für die Erstellung der elektronischen Signatur verwendet wurde, ist sichergestellt, dass die Daten zur und von der Signaturkarte nicht manipuliert wurden

Die Signierung des Datenblocks (die Erstellung der Transport-Datenstruktur) wird nicht weiter durchgeführt, wenn die Überprüfung der Signatur fehlschlägt. Dieses wird dem Benutzer angezeigt.

4.1.5 SF5 - Behandlung von OCSP Daten zur Zertifikatsvalidierung

Die SAK ermöglicht es, OCSP Informationen in die Prüfung eines Zertifikats mit einzubeziehen. Der Zugriff auf den OSCP Dienst erfolgt über die Applikation. Die SAK erstellt die Anfrage und übergibt diese der Anwendung mit der zugehörigen URL. Die Anwendung führt die konkrete Anfrage durch und stellt die OCSP Antwort bereit. Damit die SAK eine OCSP Antwort mit einbeziehen kann, muss die elektronische Signatur der Antwort verifiziert werden.

Die OCSP Antwort wird von einem OCSP Dienst bereitgestellt bzw. steht zwischengespeichert schon zur Verfügung. Eine OCSP Antwort zu einem bereits geprüften Zertifikat wird über die

[4] Der verwendete Algorithmus wird über das Konfigurationsfile, welches eine Aufstellung über die aktuell gültigen Algorithmen und Parameter beinhaltet, festgelegt.

Lebenszeit der Instanz der SAK bis maximal zu der in der Antwort hinterlegten Gültigkeitszeit der Antwort vorrätig gehalten. Der OCSP Anfrage wird ein Zufallswert mitgegeben, der in der zugehörigen Antwort zurückgeliefert wird. Stimmen die Werte nicht überein, wird die Antwort verworfen. Die Gültigkeit der Antwort wird durch Verifikation der Signatur sichergestellt. Die Verifikation besteht aus der mathematischen Prüfung der Signatur der OSCP Antwort gegen das in der Antwort enthaltene Responderzertifikat. Zu dem Responderzertifikat muss eine Zertifikatskette erstellt werden können, deren Wurzelzertifikat mit einem der aus der Konfigurationsdatei bekannten Wurzelzertifikate identisch ist. Zur Erstellung der Zertifikatskette verwendet die SAK die Zertifikate aus der Konfigurationsdatei.

Die Signatur der OCSP Antwort basiert auf den Algorithmen SHA1, SHA-224, SHA-256, SHA-384, SHA-512 oder RIPEMD-160 und RSA.

Falls die SAK nicht in der Lage sein sollte die Signatur zu verifizieren, wird die OCSP Antwort nicht importiert und eine entsprechende Meldung ausgegeben. Eine gültige OCSP Antwort kann von der SAK zur weiteren Verwendung zwischengespeichert werden.

Wenn die SAK die OCSP Information zur Verifikation eines Zertifikats benutzt, wird die OCSP Signatur geprüft. Die Signatur basiert auf o.g. Hashes. Für die Verifikation wird der Hash der signierten Daten berechnet, der Original-Hash aus der Signatur mit dem RSA-Algorithmus extrahiert und der öffentliche Schlüssel aus dem OCSP Signierzertifikat verwendet. Anschließend wird noch das Zertifikat des Erstellers der OCSP Antwort nach RSA verifiziert.

Nach Prüfung der elektronischen Signatur wird eine entsprechende Meldung über die Gültigkeit der OCSP Antwort im Rahmen der Signaturverifikation (SF2) angezeigt. Weiterhin kann der Inhalt der OCSP Antwort mit dem entsprechenden Zertifikat angezeigt werden.

Die URL für die OCSP Anfrage wird über die in der Konfiguration hinterlegte Zuordnung zum Ausstellerzertifikat ermittelt.

4.1.6 SF6 - Anbringung von Zeitstempeln

Die SAK ermöglicht den Einsatz von Zeitstempeln. Hierzu wird ein externer Zeitstempeldienst verwendet. Der Zugriff auf den Zeitstempeldienst erfolgt über die Applikation. Die SAK generiert die Anfrage und übergibt diese der Applikation. Diese entscheidet darüber, welcher konkrete Zeitstempeldienst in Anspruch genommen wird und stellt die Antwort bereit. Die Korrektheit des Zeitstempels wird durch die SAK sichergestellt.

Die SAK verifiziert die elektronische Signatur des Zeitstempels mit dem öffentlichen Schlüssel des Zertifikats, das benutzt wurde, den Zeitstempel zu signieren. Die Verifikation besteht aus der mathematischen Prüfung der Signatur der Antwort des Zeitstempeldienstanbieters (TSP) gegen das in der Antwort enthaltene Responderzertifikat.

Das Responderzertifikat muss gültig sein. Die Signaturprüfung erfolgt durch Vergleich des berechneten Hashwerts der Signatur, der auf SHA-1, SHA-224, SHA-256 SHA-384, SHA-512 oder RIPE-MD 160 basiert, mit dem Original-Hashwert, der mit dem öffentlichen Schlüssel des Zeitstempel-Signierzertifikats nach dem RSA-Algorithmus aus dem Signaturwert extrahiert wird.

Antworten zu Zeitstempelanfragen werden nur dann akzeptiert, wenn diese mit einem Zertifikat signiert wurden zu dem eine Zertifikatskette erstellt werden kann, die der Validierung nach RFC 3280 standhält. Zur Erstellung der Zertifikatskette verwendet die SAK ausschließlich Zertifikate der in der Konfigurationsdatei enthaltenen Zertifikatstämme.

Nachdem der Zeitstempel importiert wurde, wird dieser in die Daten integriert.

4.1.7 SF7 - Validierung von Zeitstempeln

Die SAK verifiziert die elektronische Signatur des Zeitstempels mit dem öffentlichen Schlüssel aus dem Zertifikat mit dem der Zeitstempel signiert wurde. Das Zertifikat, mit dem der Zeitstempel signiert wurde muss für die SAK verifizierbar sein. Er ist dann verifizierbar, wenn zu ihm eine Zertifikatskette erstellt werden kann, die der Validierung nach RFC 3280 standhält. Zur Erstellung der Zertifikatskette verwendet die SAK ausschließlich Zertifikate der in der Konfigurationsdatei enthaltenen Zertifikatstämme.

Die Signatur kann auf SHA-1, SHA-224, SHA-256 SHA-384, SHA-512 oder RIPE-MD 160 Hashwerten beruhen. Für die Verifizierung wird der Hashwert berechnet und mit dem Original-Hash verglichen, der aus dem Signaturwert mit dem öffentlichen Schlüssel des Erstellerzertifikats des Zeitstempels nach dem RSA-Algorithmus und dem öffentlichen Schlüssel des Erstellerzertifikats des Zeitstempels extrahiert wurde.

Das Ergebnis der Zeitstempelprüfung und die im Zeitstempel enthaltenen Informationen werden dem Anwender angezeigt. Weiterhin kann das Zertifikat, mit dem der Zeitstempel signiert wurde, angezeigt werden.

4.1.8 SF8 - Verwaltung der SFs mit der Konfigurationsdatei

Variable Programmkonfigurationen werden anhand einer signierten Konfigurationsdatei bereitgestellt. Die Konfigurationsdatei umfasst die Einstellungen zur Sicherheitseignung von Parametern. Die SAK sieht eine Aktualisierung der Konfigurationsdatei vor. Die Konfigurationsdatei besitzt eine festgelegte Gültigkeitsdauer, nach deren Ablauf sie nicht mehr geladen wird und die SAK mit einer entsprechenden Meldung den Start verweigert.

Die Konfigurationsdatei wird bei jedem Programmstart geladen und anhand einer digitalen Signatur verifiziert. Falls keine Konfigurationsdatei gefunden wird, wird der Anwender darauf hingewiesen, ein solches zur Verfügung zu stellen.

Die Signatur der Konfigurationsdatei wird von der SAK verifiziert und nur bei Korrektheit wird das Konfigurationsfile geladen und die Parameter werden übernommen und angewandt.

Die folgenden Parameter werden durch das Konfigurationsfile festgelegt:

- Gültigkeit/Ablaufdatum des Konfigurationsfiles
  Nach diesem Datum wird das Konfigurationsfile vom Programm nicht mehr geladen und muss durch ein aktuelleres ersetzt werden.

- Tabelle mit den gültigen Hash-Algorithmen und deren Laufzeit bzw. Ablaufdatum ihrer Sicherheitseignung[5].
- Tabelle mit den gültigen Verschlüsselungs-Algorithmen (RSA), Parametern (Schlüssellänge) und dem Ablaufdatum ihrer Sicherheitseignung[5].
- Zertifikatsstämme bekannter Zertifikatsaussteller in Dateiform.
- Tabelle mit einer Zuordnung von Zertifikatsstämmen zu Responder-URLs für die OCSP Abfrage.
- Tabelle mit sprachspezifischen Strings für andere Sprachen außer de-DE.
- Tabelle mit ergänzenden OIDs, deren Kurz- und Langname, soweit nicht bereits in den Bibliotheken hinterlegt.
- Tabelle mit XML-IDs für XML-Signature. Da die SAK ohne die Kenntnis von Schemadateien arbeitet, wird die Festlegung weiterer ID-Attribute und ihrer URI benötigt. Diese werden ggf. hier über die Konfiguration ergänzt.

4.1.9 SF9 - Schutz vor Manipulation der Programmdatei

Die Programmdatei wird vom Hersteller signiert. Das Programm verfügt über den zugehörigen öffentlichen Schlüssel und verifiziert beim Programmstart die mathematische Korrektheit der Signatur. Schlägt die Verifikation fehl, wird der Anwender auf diesen Umstand hingewiesen und die SAK verweigert die weitere Ausführung.

4.2 Benutzerschnittstelle

Im Folgenden finden Sie die Anzeigen und Bedienelemente des Infotech Signer. Die Gestaltung und Farbgebung der Dialoge ist von dem verwendeten Betriebssystem und Oberflächeneinstellungen abhängig. Die folgenden Screenshots wurden unter Windows 2000 mit Standarddialogeinstellungen vorgenommen.

In diesem Abschnitt wird die reguläre Arbeitsweise und die zugehörigen Anzeigen dargestellt. Zu ggf. auftretenden Fehlersituationen und deren Handhabung finden Sie nähere Hinweise im Abschnitt 4.3, Sicherheitshinweise und -empfehlungen.

4.2.1 TrayIcon

Die SAK erstellt zu Programmstart ein Trayicon, welches während der gesamten Programmlaufzeit bestehen bleibt:

[5] Die Verwendung von Hash-Algorithmen sowie von Verschlüsselungsalgorithmen und –parametern zur Signaturausführung wird nach Ablauf ihrer Sicherheitseignung nicht unterbunden, jedoch dem Benutzer durch eine eindeutige Warnmeldung zur Kenntnis gebracht (FDP_DAU.2).

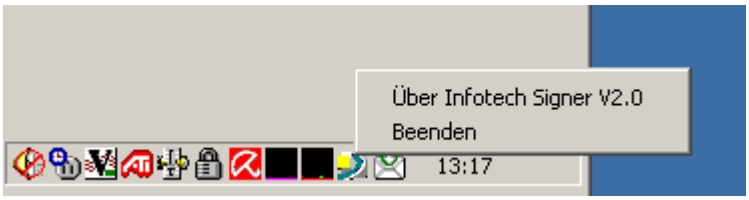
Abbildung 4.1 - Screenshot TrayIcon

Zu dem TrayIcon gehört ein TrayMenu. Durch Klick mit der rechten Maustaste auf das TrayIcon wird die Anzeige des TrayMenüs ausgelöst. Das TrayMenü enthält Einträge zur Anzeige der Statusanzeige und des Informationsdialogs:

Abbildung 4.2 - Screenshot TrayMenü

Die SAK kann in den regulären Arbeitszuständen nur von der ihn verwendenden Applikation beendet werden. In besonderen Fällen, unter Abbruch der Verbindung mit der Applikation, kann die SAK manuell durch den Benutzer beendet werden. Ist dies der Fall, wird das Menü um den Menüpunkt „Beenden“ erweitert, dessen Betätigung zur Beendigung der SAK führt:

Abbildung 4.3 - Screenshot TrayMenü - erweitert

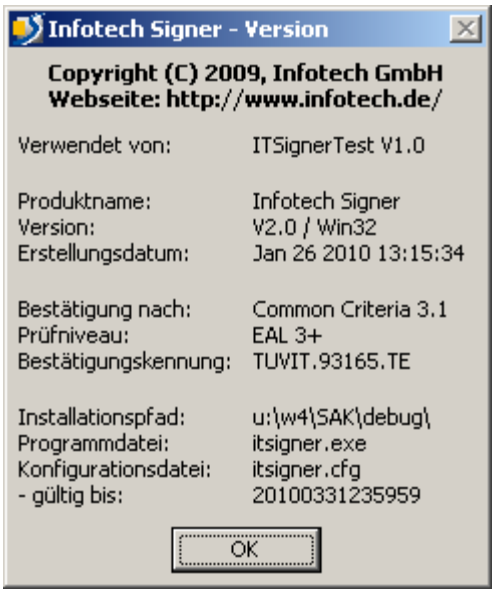


Durch Klick mit der linken Maustaste auf das TrayIcon oder durch Aktivierung des Menüpunkts „Statusanzeige“ im TrayMenü wird die Statusanzeige aktiviert und im Vordergrund dargestellt.

4.2.2 Informationsdialog

Die Anzeige des Informationsdialogs erfolgt durch Aktivierung des Traymenüpunktes „Über Infotech Signer V2.0“ und führt zur Ausgabe der Versionsinformationen, des Speicherortes der Programmdatei der SAK und des in der Konfigurationsdatei hinterlegten Ablaufdatums:

Abbildung 4.4 - Screenshot Informationsdialog



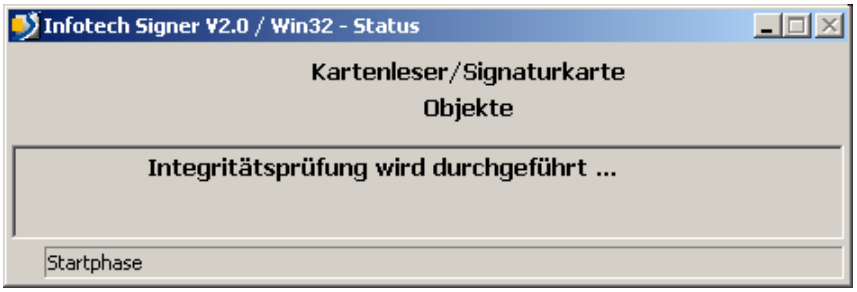
4.2.3 Statusanzeige

Die Statusanzeige besteht aus einer Gruppe von Anzeigeelementen, die über den Zustand der SAK Auskunft erteilen. Die Gruppe besteht aus einer Liste der Kartenlesegeräte, der Ablagefächer, einer Dialogfläche und einem Statusbalken mit einer kleinen Darstellungsfläche für ein weiteres Icon an der linken Seite.

Während der Startphase sind noch keine Kartenlesegeräte bekannt und alle Ablagefächer leer. Die Dialogfläche dient zur Informationsausgabe und bei Bedarf zur Entgegennahme von Benutzerentscheiden.

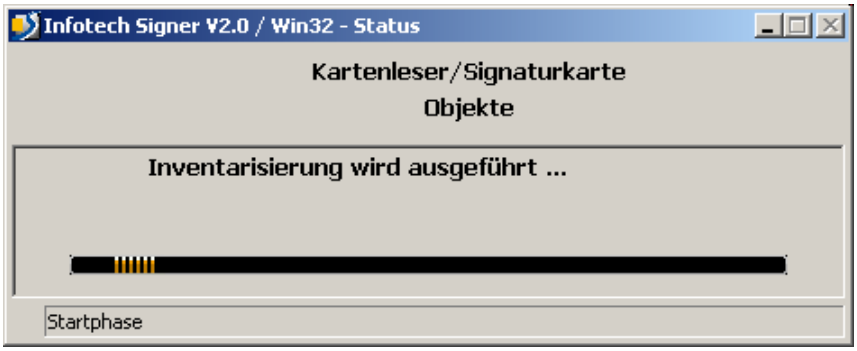
Der Statusbalken zeigt den aktuellen Gesamtzustand der SAK, hier unmittelbar nach Programmstart:

Abbildung 4.5 - Screenshot Statusanzeige - Initialisierung



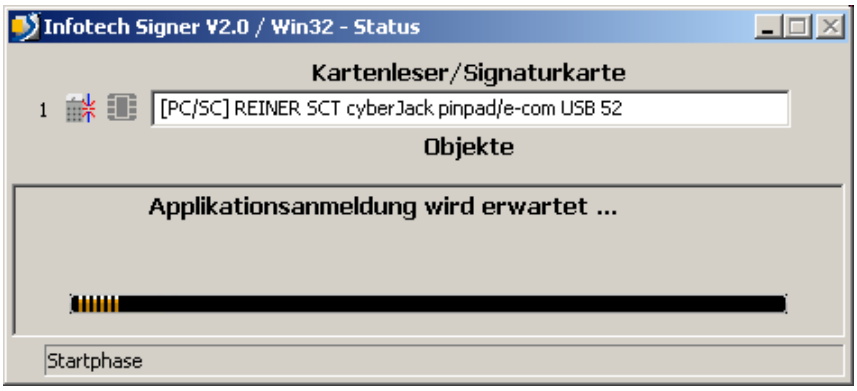
Nach Abschluss der Startphase erfolgt die Inventarisierung, bei der die in der Konfigurationsdatei enthaltenen Informationen weiterverarbeitet und die verfügbaren Kartenlesegeräte ermittelt werden:

Abbildung 4.6 - Screenshot Statusanzeige - Inventarisierung



Nach durchgeführter Inventarisierung sind die angeschlossenen Kartenlesegeräte bekannt und ihre Namen und Zustände werden in der Liste der Kartenlesegeräte angezeigt. Anschließend wird die Kommunikation mit der Applikation aufgenommen und deren Anmeldung erwartet.

Abbildung 4.7 - Screenshot Statusanzeige - Applikationsanmeldung



Die Anzeige stellt ein gefundenes Kartenlesegerät dar. Der Treiber zu dem Kartenlesegerät wurde noch nicht geladen, eine möglicherweise vorhandene Signaturkarte wurde noch nicht bestimmt. Der Name des Kartenlesergeräts wird in dem angezeigten Eingabefeld dargestellt.

In der Liste der Kartenlesegeräte können simultan bis zu 3 Kartenlesegeräte angezeigt werden. Falls mehr Kartenlesegeräte erkannt werden, erhält die Liste Scrollelemente (und) zur Navigation.

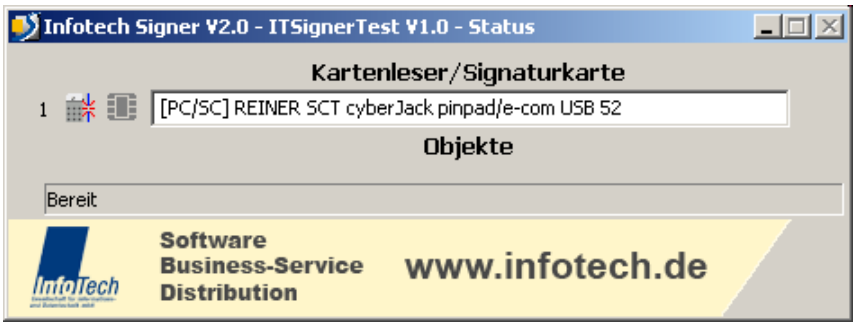
Jeder Listeneintrag besteht aus:

- Einer laufenden Nummer, beginnend mit 1

- Symbol für den Zustand von Kartenleser / Treiber, entsprechend READER.STATUS
  - 0 Undefiniert — Kartenlesertreiber wurde noch nicht geladen
  - 1 Bereit — Treiber geladen, Kartenleser verbunden
  - 2 Fehler — Fehlerzustand im Treiber aufgetreten
- Symbol für den Zustand der Signaturkarte, nach CARD.STATUS
  - 0 Undefiniert — Der Zustand wurde noch nicht bestimmt
  - 1 SingleClose — Einfachsignaturkarte, geschlossen
  - 2 SingleOpen — Einfachsignaturkarte, geöffnet
  - 3 MultiClose — Mehrfachsignaturkarte, geschlossen
  - 4 MultiOpen — Mehrfachsignaturkarte, geöffnet
  - 5 Fehler — Fehlerzustand eingetreten
- Der Anzeige des Kartenlesergerätenamens, wie er an die Applikation weitergegeben und von ihr zur Benennung des für einen Signaturvorgang zu verwendenden Kartenlesegeräts verwendet wird. Der Hintergrund der Anzeige wird in Abhängigkeit von der Verwendung der Signaturkarte und ihrem Zustand eingefärbt.
  - Weiß — Leerlaufzustand, Signaturkarte geschlossen
  - Cyan — Kartenlesegerät zur Signaturausführung ausgewählt
  - Gelb — Signaturkarte zur Signaturausführung geöffnet
  - Rot — Leerlaufzustand, Signaturkarte geöffnet

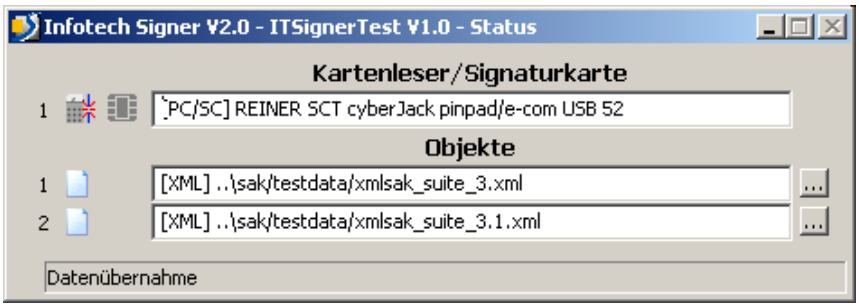
Abhängig von der angemeldeten Applikation wird eine Grafik unterhalb der Titelleiste, unterhalb der Statuszeile, am linken oder am rechten Fensterrahmen integriert. Die Grafik enthält keine aktiven Elemente und bleibt während der Programmlaufzeit bestehen.

Abbildung 4.8 - Screenshot Statusanzeige - Applikationsgrafik



Zur Durchführung oder Verifikation von Signaturen übergibt die Applikation einen oder mehrere Datenblöcke mit Meta- und Parameterdaten. Die Datenblöcke werden mit ihren Begleitdaten in Ablagefächer hinterlegt und die Metadaten zusammen mit den Bearbeitungszuständen in der Liste der Ablagefächer angezeigt.

Abbildung 4.9 - Screenshot Statusanzeige - Datenblöcke geladen



In der Liste der Ablagefächer können simultan bis zu 5 Ablagefächer dargestellt werden. Falls mehr als 5 Ablagefächer verwendet werden, erhält die Liste Scrollelemente (und) zur Navigation.

Jeder Listeneintrag besteht aus

- Einer laufenden Nummer, beginnend mit 1
- Symbol für den Verarbeitungszustand in Abhängigkeit vom geladenen Datenblock
  Bei Dokumenten nach DOCUMENT.STATUS

	0	unbearbeitet	Dokument ist unbearbeitet
	1	wird signiert	Dokument wird gerade signiert
	2	wird verifiziert	Dokument wird gerade verifiziert
	3	signiert	Dokument ist signiert
	4	geprüft	Dokument ist verifiziert
	5	signiert und geprüft	Dokument ist signiert und verifiziert
	6	ungültig	Dokument ist ungültig

  Bei Zertifikaten nach CERTIFICATE.STATUS (nur Verifikation)

	0	undefiniert	Zertifikat ist unbearbeitet
	1	gültig	Zertifikat ist gültig
	2	ungültig	Zertifikat ist ungültig

- Symbol für das Prüfergebnis in Abhängigkeit vom geladenen Datenblock
  Bei Dokumenten nach VERIFICATION.STATUS

	0	undefiniert	Dokument ist ungeprüft
	1	gültig	Dokument ist geprüft und gültig
	2	ungültig	Dokument ist geprüft und ungültig
	11	bestreitbar	Dokument ist unvollständig validiert

  Bei Zertifikaten nach CERTOCSP.STATUS

	0	ungeprüft	Es ist noch keine OCSP Prüfung erfolgt
	1	gut	Zertifikat ist beim Aussteller als gültig bekannt
	2	zurückgerufen	Zertifikat wurde beim Aussteller zurückgerufen
	3	unbekannt	Zertifikat ist beim angefragten Aussteller nicht bekannt

?	4	abfragefehler	Es konnte keine gültige Antwort eingeholt werden
?	5	aussteller unbekannt	Der Aussteller des Zertifikats ist nicht bekannt
?	6	verifikationsfehler	Fehler bei Verifikation des Responderzertifikats

- Der Anzeige der Metadaten zu dem im Ablagefach befindlichen Datenblock
- Dem Navigationselement … zur Anzeige der Detailinformation zu den im Ablagefach befindlichen Daten (siehe 4.2.5, Detailanzeige).

4.2.3.1 Statusanzeige bei Signaturausführung

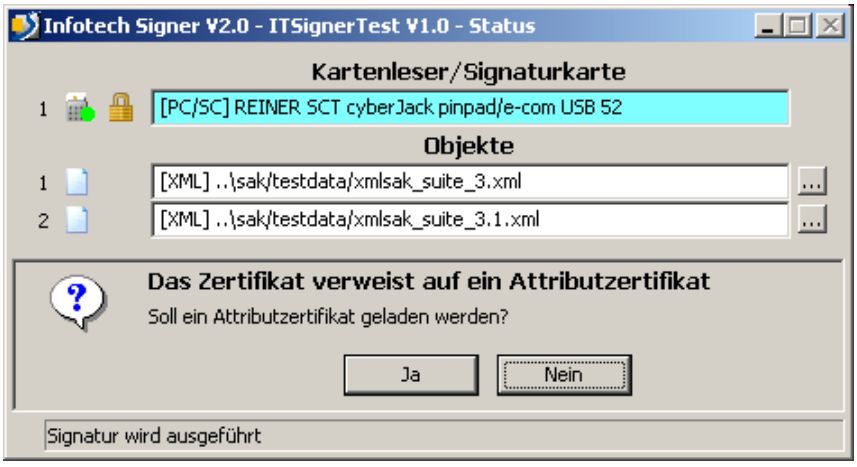
Zur Durchführung von Signaturaufträgen wird das zu verwendende Kartenlesegerät von der Applikation benannt.

Sofern die zur Signaturausführung erforderlichen Parameter vorliegen, stellt die SAK die Verbindung zum benannten Kartenlesegerät her und ermittelt die Eigenschaften der Signaturkarte. Der Eintrag des für den Signaturvorgang verwendeten Kartenlesegeräts wird durch Einfärbung des Hintergrunds hervorgehoben.

Wenn die Signaturkarte gelesen werden konnte und zur Multisignatur geeignet ist, wird dies durch den Einstelldialog zur Multisignatur angezeigt, sofern die Einstellungen nicht bereits innerhalb der letzten 5 Minuten vorgenommen wurden und die eingestellten Beschränkungen noch erfüllt sind. Der Dialog und die Einstellmöglichkeiten sind im Abschnitt 4.2.4, Multisignatureinstellungen, beschrieben.

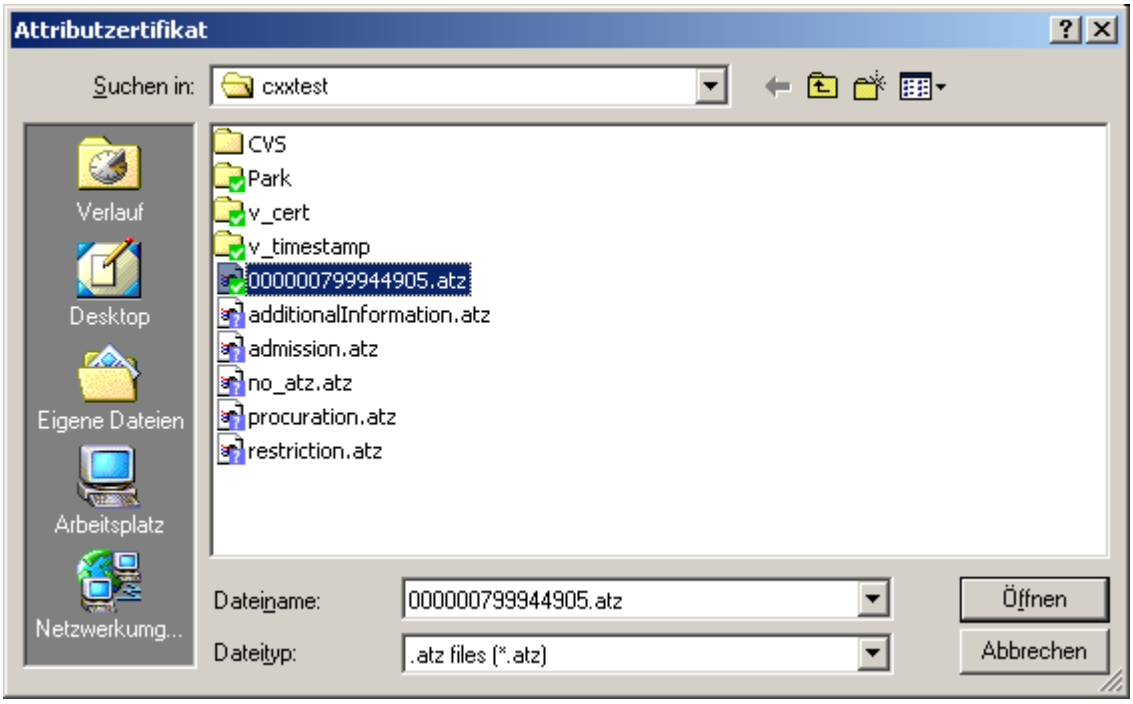
Wenn das Signaturzertifikat den Hinweis auf ein Attributzertifikat enthält, kann der Benutzer ein vorhandenes Attributzertifikat durch die Applikation hinzufügen lassen:

Abbildung 4.10 - Screenshot Statusanzeige - Attributzertifikat laden



Bei Bejahung des Dialogs fordert die SAK die Applikation zur Übermittlung eines Attributzertifikats auf. Die Applikation lässt einen entsprechenden Dialog zum Öffnen einer Datei erscheinen und übermittelt die in der Datei befindlichen Daten zur SAK.

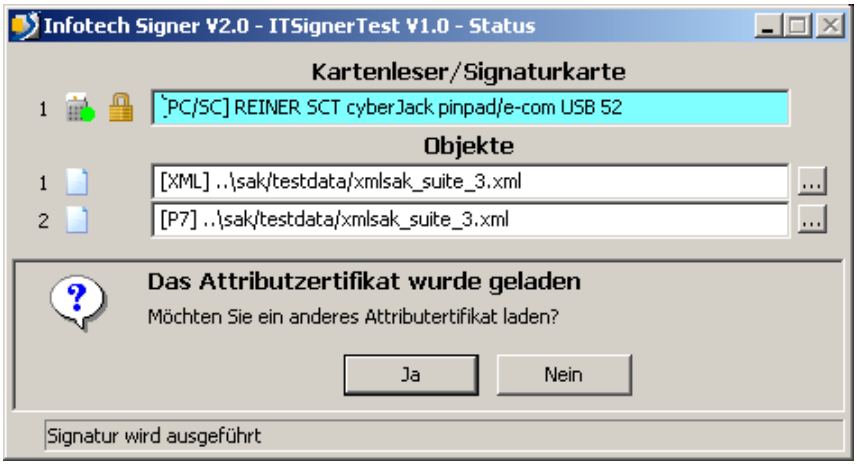
Abbildung 4.11 - Screenshot Dateiauswahldialog (Fachapplikation)



Die SAK nimmt den Dateiinhalt entgegen und prüft, ob es sich dabei um ein Attributzertifikat handelt, ob es das Signaturzertifikat referenziert und gültig (nicht abgelaufen) ist.

Wenn das Attributzertifikat den Prüfungen genügt, wird dies wie folgt angezeigt:

Abbildung 4.12 - Screenshot Statusanzeige - Attributzertifikat geladen



Zur Aufnahme des Attributzertifikats in die Signaturen ist der vorstehende Dialog zu verneinen.

Wenn das zu verwendende Zertifikat und ggf. das Attributzertifikat für den Signaturvorgang festgelegt sind, führt die SAK die Signatur für alle in den Ablagefächern befindlichen Dokumente durch. Dabei kann die Signaturkarte nicht gewechselt werden.

Zu einer Signaturausführung muss die Signaturkarte durch Benutzereingabe (PIN) geöffnet vorliegen. Wenn es sich bei der Signaturkarte um eine Einzelsignaturkarte (silbernes Schloss) handelt, ist die PIN zu jeder Signatur einzugeben. Falls es sich jedoch um eine Mehrfachsignaturkarte handelt (goldenes Schloss), kann der Zustand bereits vorliegen und die Signaturkarte wird ohne weitere Nachfrage zur Signaturausführung verwendet.

Auch Multisignaturkarten sind mindestens bei der ersten Signaturausführung durch PIN-Eingabe zu öffnen. Wenn die Signaturkarte die eingegebene PIN ablehnt, die PIN nicht innerhalb der vorgegebenen Zeit eingegeben wurde, oder andere korrigierbare Fehler auftreten, kann der Benutzer die PIN-Eingabe wiederholen.

Das Auftreten nicht korrigierbarer Fehler, wie das Entfernen der Signaturkarte, führen zum Abbruch des gesamten Signaturauftrags und zum Verwerfen aller in dem aktuellen Signaturauftrag erzeugten Signaturen.

Die folgende Anzeigesequenz zeigt die Ausführung eines Signaturauftrags:

Abbildung 4.13 - Screenshot Statusanzeige - Signaturbeginn

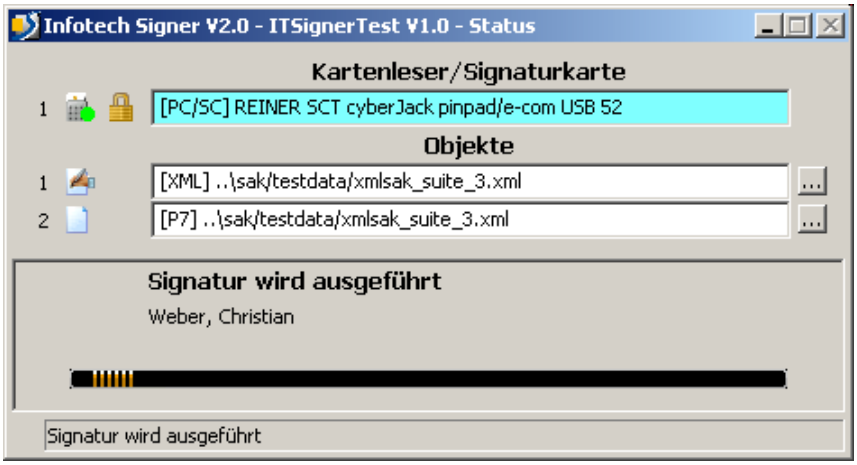


Abbildung 4.14 - Screenshot Statusanzeige - Signatur nach PIN

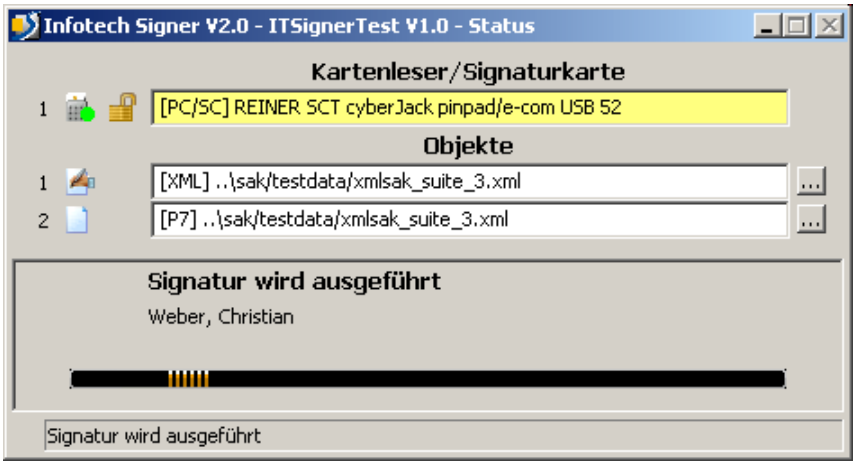
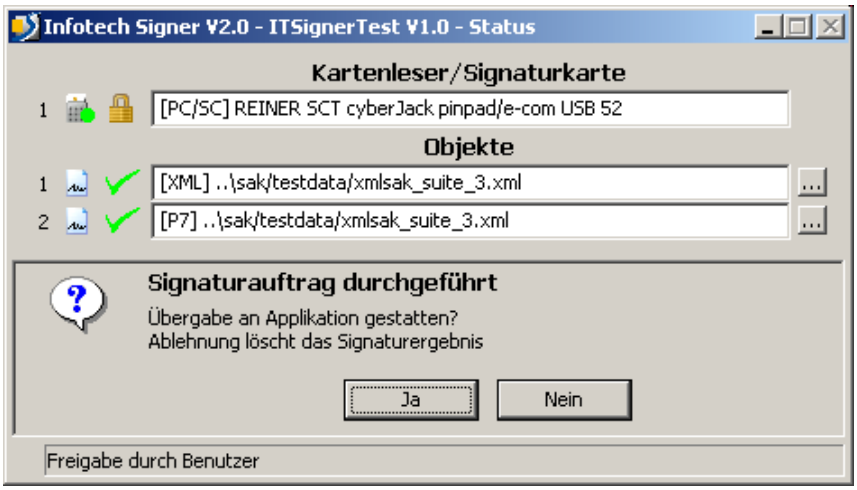


Abbildung 4.15 - Screenshot Statusanzeige - Signatur ausgeführt



Zum Abschluss des Signaturauftrags ist die Übergabe der signierten Daten an die Applikation zu bestätigen.

Der Benutzer kann Details zu den geladenen Daten zu jeder Zeit durch Anforderung der Detailanzeige einsehen. Die Detailanzeige ist zu jeder Zeit zwischen der Datenübernahme der Rohdaten, mindestens bis zur Übergabe der Ergebnisdaten, verfügbar.

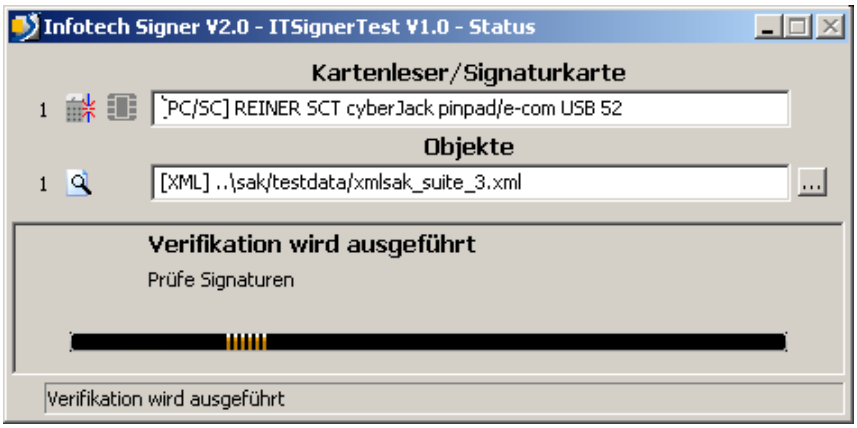
Die Details nach Signaturausführung enthalten zu jedem Dokument nur die Daten zu der gerade angefertigten Signatur, nicht zu bereits vor der Signaturausführung im Dokument vorhandenen Signaturen. Alle Signaturen können durch einen unmittelbar folgenden Verifikationsauftrag untersucht werden, soweit dies durch die Applikation unterstützt wird.

4.2.3.2 Statusanzeige bei Verifikationsausführung

Zur Durchführung von Verifikationsaufträgen ist kein Kartenlesegerät erforderlich. Die angeschlossenen Kartenlesegeräte verbleiben in ihrem Zustand, es findet kein Datenaustausch mit einer Signaturkarte statt.

Die Applikation belässt vorhandene, gerade signierte Dokumente in den Ablagefächern oder überträgt zu prüfende Dokumente oder Zertifikate und leitet dann den Auftrag ein. Die SAK informiert den Benutzer über die Durchführung des Auftrags durch folgende Anzeige:

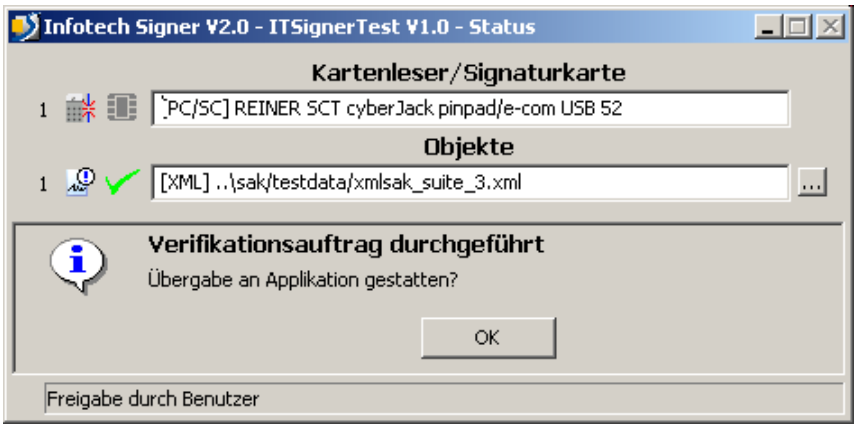
Abbildung 4.16 - Screenshot Statusanzeige - Verifikationsbeginn



Zu Beginn der Verifikation werden zunächst alle Signaturen ermittelt und anschließend verifiziert.

Nach Abschluss des Verifikationsauftrags ist die Übertragung des Ergebnisses an die Applikation zu bestätigen.

Abbildung 4.17 - Screenshot Statusanzeige - Verifikation durchgeführt



In diesem Zustand kann der Benutzer die Details zur Verifikation einsehen, weitere Prüfungen und auch Exporte vornehmen. Die Detaildaten stehen in demselben Umfang wie nach einer Signaturausführung, jedoch davon abweichend nicht nur für die angefertigte Signatur, sondern für alle ermittelten Signaturen zur Verfügung

4.2.3.3 Prüfergebnis und Prüfzeitpunkt

Nach einer Signaturausführung wird das Ergebisdokument verifiziert. Die Verifikation erstreckt sich nur auf gerade ausgeführte Signatur. Bei Verifikationsaufträgen werden alle Signaturen und Zertifikate der vorgelegten Prüfobjekte (Dokumente und / oder Zertifikate) geprüft. In der Statusanzeige wird für jedes Prüfobjekt das Symbol des schlechtesten Prüfergebnisses aller durchgeführten Prüfmethoden (Validierung, OCSP-Prüfung) angezeigt.

Wenn noch keine Prüfung durchgeführt wurde, wird dies mit dem leeren Platzhalter angezeigt. Nur wenn bei keiner der durchgeführten Prüfungen ein Fehler erkannt wurde, kann die Signatur oder das Zertifikat gültig sein. Wenn eine der Prüfungen Inkonsistenzen aufweist (z.B. weil ein Dokument nach der Signatur verändert wurde oder das Zertifikat zum Prüfzeitpunkt zurückgerufen war) wird dies mit dem Symbol angezeigt. Ergeben sich bei der Prüfungen Warnungen, jedoch keine Fehler, wird dies mit symbolisiert. In diesem Fall konnte die Güligkeit einer Signatur oder eines Zertifikats nicht vollständig bestimmt werden (z.B. weil das Signaturzertifikat nicht mitsigniert wurde oder beim Zertifikatsausstelle nicht bekannt ist). Die Gültigkeit eines Dokuments kann dann ggf. nur aus weiteren Informationen (etwa dem Signaturzeitpunkt einer Archivierung) ermittelt werden. In jedem Fall sollten die Detailinformatinen zu dem Prüfobjekt (Zertifikate und Signaturen von Dokumenten) eingesehen und zur Bewertung herangezogen werden (siehe Abschnitt 4.2.5, Detailanzeige).

Die Gültigkeit von Signaturen und Zertifikaten werden zu Prüfzeitpunkten festgestellt. Dokumente tragen i.d.R. ihren Prüfzeitpunkt in Form eines Zeitstempels oder einer mitsignierten Systemzeit (die Signaturzeit). Fehlt dieser Prüfzeitpunkt im Dokument, oder wurde ein Zertifikat zur Prüfung vorgelegt, wird die aktuelle Systemzeit als Prüfzeitpunkt verwendet. Der Prüfzeitpunkt wird während der Prüfung bestimmt und die Zeitquelle bei Signaturen in deren Detailanzeigen ausgewiesen (siehe Abschnitt 4.2.5.3, Detailanzeige (Signatur)). Er kann nicht durch die Fachapplikation oder Benutzerangaben beeinflusst werden. Für Zertifikate erhält man die Gültigkeitsinformation stets zur aktuellen Systemzeit.

Die Prüfergebnisse werden anhand der durchgeführten Prüfungen in der Statusanzeige verdichtet am Prüfobjekt angezeigt. Wenn eine Prüfmethode wie die OCSP Prüfung noch nicht angewandt wurde, wird sie in das Ergebnis nicht einbezogen. Durch die Validierung der Objekte zusammen mit der optionalen Einbeziehung der Zustandsinformation ihrer Zertifikate vom ihren Ausstellern erhält man eine umfassende Aussage über die Gültigkeit der Signaturen.

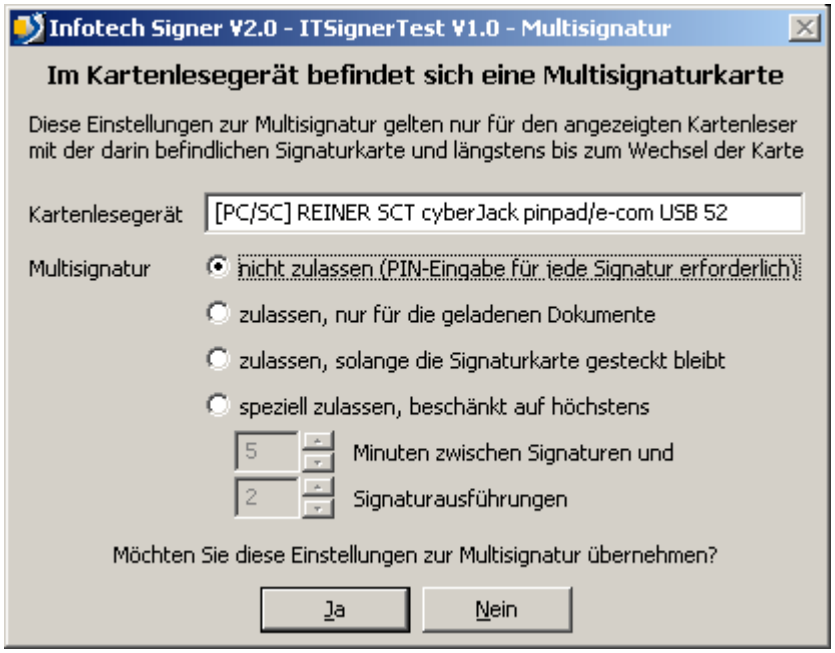
4.2.4 Multisignatureinstellungen

Der Dialog zu Multisignatureinstellungen wird nach der Prüfung des Kartenlesegeräts und der Signaturkarte vor der Signaturausführung angezeigt, wenn:

- das Kartenlesegerät verfügbar ist und eine Multisignaturkarte enthält
  – und –
- in dem aktuellen Signaturvorgang mehr als die nach der wirksamen Einstellung zugelassene Anzahl von Signaturen zu leisten sind
  – oder –
  die letzte Signatur mit diesem Kartenlesegerät unter Verwendung derselben Signaturkarte durchgeführt und innerhalb der vergangenen 5 Minuten in dem Dialog nicht der Einzelsignaturmodus gewählt wurde (Option: Multisignatur nicht zulassen).

Der Dialog gestattet die Änderung der Handhabung von Multisignaturkarten. Voreingestellt ist die Nutzung von Multisignaturkarten als Einzelsignaturkarte. D.h. für jede Signaturausführung ist die Eingabe der PIN zur Authentisierung des Benutzers gegen die Signaturkarte erforderlich und die Multisignaturkarte wird nach jeder Signatur geschlossen. Dieses Verhalten ermöglicht anderen Applikationen zwischen den Signaturausführungen über dasselbe Kartenlesegerät auf dieselbe Signaturkarte zuzugreifen.

Abbildung 4.18 - Screenshot Multisignatureinstellungen


Infotech Signer V2.0 - ITSignerTest V1.0 - Multisignatur
Im Kartenlesegerät befindet sich eine Multisignaturkarte
Diese Einstellungen zur Multisignatur gelten nur für den angezeigten Kartenleser mit der darin befindlichen Signaturkarte und längstens bis zum Wechsel der Karte
Kartenlesegerät
[PC/SC] REINER SCT cyberJack pinpad/e-com USB 52
Multisignatur
nicht zulassen (PIN-Eingabe für jede Signatur erforderlich)
zulassen, nur für die geladenen Dokumente
zulassen, solange die Signaturkarte gesteckt bleibt
speziell zulassen, beschänkt auf höchstens
5
Minuten zwischen Signaturen und
2
Signaturausführungen
Möchten Sie diese Einstellungen zur Multisignatur übernehmen?
Ja
Nein

Der Dialog zeigt den Namen des zur Signaturausführung festgelegten Kartenlesegeräts und einige Einstellmöglichkeiten zur Nutzung der Multisignaturkarte.

Die Optionen bedeuten im Einzelnen

- „Multisignatur nicht zulassen (PIN-Eingabe für jede Signatur erforderlich)“
  Das Setzen dieser Option entspricht der Voreinstellung. Zwischen Signaturausführungen wird die Signaturkarte geschlossen und steht anderen Applikationen zur Verfügung.
- „Multisignatur zulassen, nur für die geladenen Dokumente“
  Diese Option bewirkt das Halten der Signaturkarte in geöffnetem Zustand für die Anzahl von Signaturen, die zur Signaturausführung auf allen Ablagefächern erforderlich ist.
- „Multisignatur zulassen, solange die Signaturkarte gesteckt bleibt“
  Durch Anwahl dieser Option bleibt die Multisignaturkarte nach erfolgter Authentisierung in geöffnetem Zustand gehalten, bis sie entfernt oder das Programm beendet wird.
- „Multisignatur speziell zulassen, beschränkt auf...“
  Diese Option gestattet die Einstellung spezieller Beschränkungen der Nutzung der Signaturkarte. Die Karte bleibt geöffnet, bis die eingestellte Anzahl zugelassener Signaturausführungen erreicht oder zwischen zwei Signaturen mehr als die eingestellte Zeit verstrichen ist. Wird innerhalb der eingestellten Zeit erneut eine Signatur durchgeführt, beginnt die Zeitzählung erneut. Die Zeitangabe erfolgt in ganzen Minuten ohne Nachkommastellen. Der Wert ‚0‘ deaktiviert die Zeitbeschränkung, d.h. die Multisignaturkarte bleibt zeitlich unbeschränkt geöffnet.

Die Anwahl einer der letzten beiden Optionen versetzt die SAK in einen nicht signaturgesetzkonformen Betriebszustand, weil die zur Signatur vorgelegten Dokumente vor der Signatur ggf. nicht eingesehen werden können. Für signaturgesetzkonformen Betrieb dürfen sie nicht verwendet werden. Der Betrieb der SAK in diesem Zustand ist nicht Gegenstand der Evaluierung.

Das Feld zur Einstellung der zugelassenen Signaturausführungen ist mit der Anzahl befüllter Ablagefächer vorbelegt. Die Einstellung von „0“ bedeutet, dass die Anzahl der Signaturausführung nicht beschränkt ist. Das Feld hat einen Wertebereich von „0“ bis „999“.

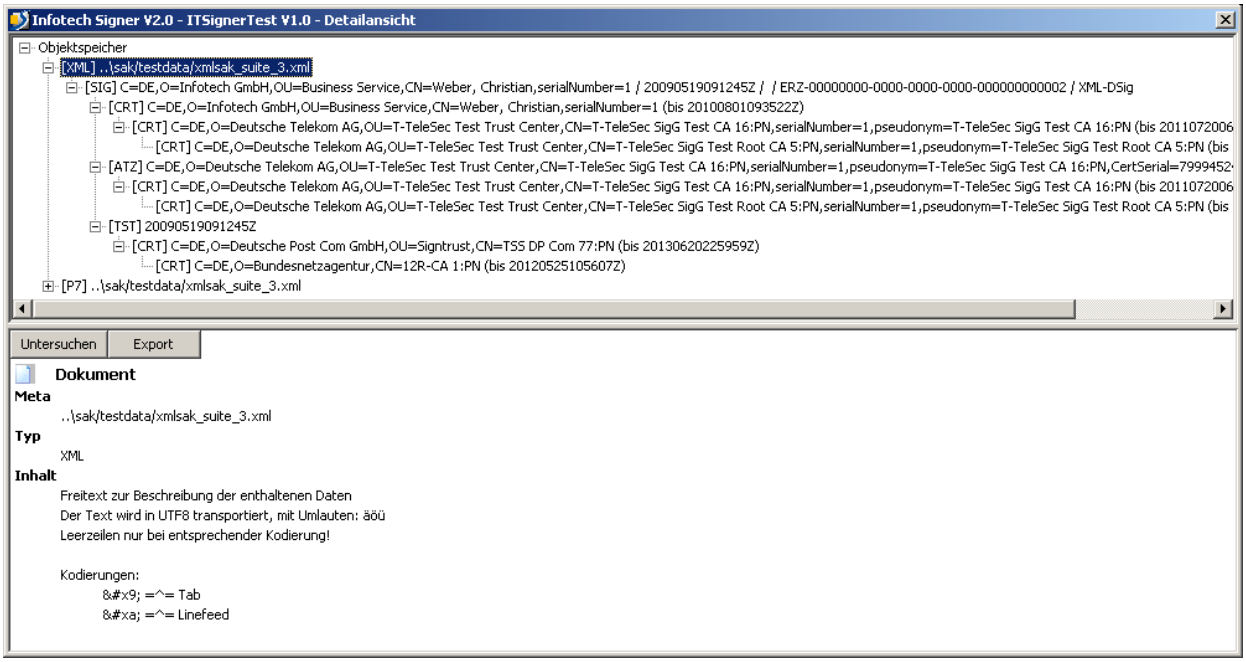
Das Feld zur Einstellung der Zeit zwischen den Signaturen beschränkt die zugelassene Leerlaufzeit der geöffneten Signaturkarte. Wird die Karte über ein größeres als das eingestellte Zeitintervall nicht verwendet, wird sie geschlossen. Die Einstellung von „0“ bedeutet, dass die Leerlaufzeit nicht beschränkt ist. Das Feld hat einen Wertebereich von „0“ bis „30“.

Bei Bejahung des Dialogs werden die getätigten Einstellungen wirksam. Wird er verneint, gilt die Voreinstellung bis der laufende Signaturvorgang abgeschlossen ist, jedoch wird die wiederholte Anzeige des Dialogs ermöglicht.

4.2.5 Detailanzeige

Die Detailanzeige besteht aus einer Gruppe von Anzeigeelementen, die wiederum in zwei Darstellungsformen angezeigt werden: Dem Navigationsbaum im oberen Fensterbereich und der Detailanzeige im unteren Fensterbereich. Die in der Detailanzeige dargestellte Information bezieht sich auf das im Navigationsbaum ausgewählte Element.

Abbildung 4.19 - Screenshot Detailanzeige


Infotech Signer V2.0 - ITSignerTest V1.0 - Detailansicht
Objektspeicher
[XML] ..\sak/testdata/xmlsak_suite_3.xml
[SIG] C=DE,O=Infotech GmbH,OU=Business Service,CN=Weber, Christian,serialNumber=1 / 200905190912457 / / ERZ-00000000-0000-0000-0000-000000000002 / XML-DSig
[CRT] C=DE,O=Infotech GmbH,OU=Business Service,CN=Weber, Christian,serialNumber=1 (bis 201008010935222)
[CRT] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test CA 16:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test CA 16:PN (bis 2011072006
[CRT] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test Root CA 5:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test Root CA 5:PN (bis
[ATZ] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test CA 16:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test CA 16:PN,CertSerial=7999452
[CRT] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test CA 16:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test CA 16:PN (bis 2011072006
[CRT] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test Root CA 5:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test Root CA 5:PN (bis
[TST] 200905190912457
[CRT] C=DE,O=Deutsche Post Com GmbH,OU=Signtrust,CN=TSS DP Com 77:PN (bis 201306202259597)
[CRT] C=DE,O=Bundesnetzagentur,CN=12R-CA 1:PN (bis 201205251056077)
[P7] ..\sak/testdata/xmlsak_suite_3.xml
Untersuchen
Export
Dokument
Meta
..\sak/testdata/xmlsak_suite_3.xml
Typ
XML
Inhalt
Freitext zur Beschreibung der enthaltenen Daten
Der Text wird in UTF8 transportiert, mit Umlauten: äöü
Leerzeilen nur bei entsprechender Kodierung!
Kodierungen:
	 =^= Tab

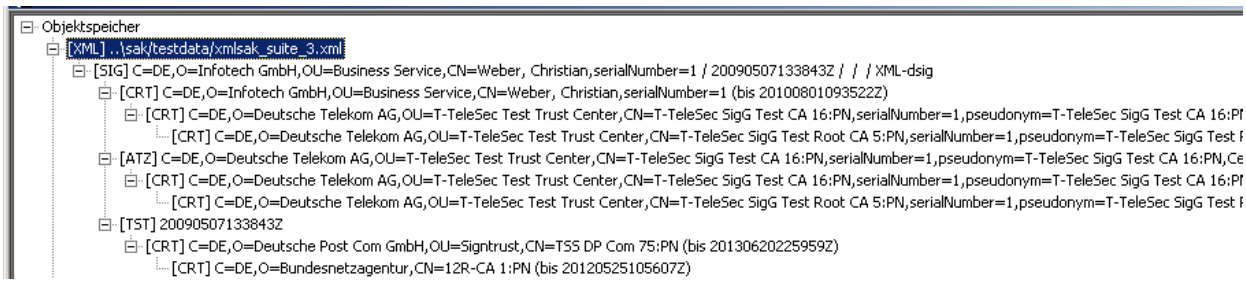
 =^= Linefeed

Die in den Detailanzeigen angebrachten Schaltflächen für Benutzeraktionen (hier: “Untersuchen“ und “Export“) sind nur nach ausgeführter Signatur oder Verifikation verwendbar. In allen anderen Zuständen führen sie zu einer Fehlermeldung.

4.2.5.1 Navigationsbaum

Der Navigationsbaum enthält diejenigen Elemente zu den Einzelauskünfte angezeigt werden können. Wurzel des Baums ist der Knoten „Objektspeicher“. Unter diesem befinden sich die in den Ablagefächern gespeicherten Objekte.

Abbildung 4.20 - Screenshot Detailanzeige - Navigationsbaum


Objektspeicher
[XML] ..\sak/testdata/xmlsak_suite_3.xml
[SIG] C=DE,O=Infotech GmbH,OU=Business Service,CN=Weber, Christian,serialNumber=1 / 200905071338437 / / / XML-dsig
[CRT] C=DE,O=Infotech GmbH,OU=Business Service,CN=Weber, Christian,serialNumber=1 (bis 201008010935222)
[CRT] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test CA 16:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test CA 16:P
[CRT] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test Root CA 5:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test
[ATZ] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test CA 16:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test CA 16:PN,Ce
[CRT] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test CA 16:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test CA 16:P
[CRT] C=DE,O=Deutsche Telekom AG,OU=T-TeleSec Test Trust Center,CN=T-TeleSec SigG Test Root CA 5:PN,serialNumber=1,pseudonym=T-TeleSec SigG Test
[TST] 200905071338437
[CRT] C=DE,O=Deutsche Post Com GmbH,OU=Signtrust,CN=TSS DP Com 75:PN (bis 201306202259597)
[CRT] C=DE,O=Bundesnetzagentur,CN=12R-CA 1:PN (bis 201205251056077)

Die Objekte werden mit einem Kürzel ihres Typs (in eckigen Klammern) und ausgewählten Kenndaten angezeigt. Mögliche Typen sind:

- [XML] oder [P7] Dokument
  Das Dokument wurde von der Applikation übergeben oder ist bereits signiert. Angezeigt werden die übergebenen Metadaten.
- [SIG] Signatur
  Angezeigt wird der X509-Name des Inhabers des verwendeten Signaturzertifikats, „/“, die zugrunde gelegte Signaturzeit, „/“, die Signaturrolle (nur bei PKCS#7-Signaturen), „/“, die Signatur ID (nur bei XML-Signaturen), „/“ und das Signaturformat („PKCS#7“ oder „XML-dsig“)
- [CRT] Zertifikat
  Der X509-Name des Inhabers und das Ende der Gültigkeit des Zertifikats werden angezeigt. Das Zertifikat wurde bei der Ausführung oder Verifikation einer Signatur oder während seiner Prüfung als Element der Zertifikatskette ermittelt.
- [ATZ] Attributzertifikat
  Angezeigt werden der Holder des Zertifikats und das Ende seiner Gültigkeit.
  Der Holder ist entweder der X509-Name des Zertifikatsausstellers und der Seriennummer des Zertifikats, der den Eigentümer identifiziert oder der X509-Name des Eigentümers, wie er in seinem Zertifikat eingetragen ist
- [TST] Zeitstempel
  Bei einem Zeitstempel wird lediglich der mitsignierte Zeitpunkt angezeigt.
- Vorgabe aus der Applikation
  Die Applikation kann ein beliebiges Objekt (Datenblock) mit beliebigem Typkennzeichen zur Signatur nach PKCS#7 vorlegen. Das angegebene Typkennzeichen wird zusammen mit den übergebenen Metadaten angezeigt.

Vor einem Objekteintrag wird ein Expansionssymbol angezeigt, welches bei Anzeige der abhängigen Objekte als Minuszeichen (⊟), sonst geschlossen als Pluszeichen (⊞) dargestellt wird.

4.2.5.2 Detailanzeige (Dokument)

Dieses Anzeigeelement enthält zwei Buttons für die Benutzeraktionen „Untersuchen“ und „Export“, die Gruppenbezeichnung der dargestellten Detailinformationen „Dokument“ und die Detailinformationen selbst. Die Detailanzeige ist einsehbar, sobald ein Dokument von der Applikation übergeben wurde, bis das zugeordnete Ablagefach wieder geleert wird.

Die Leerung des Ablagefachs erfolgt durch die Applikation, jedoch frühestens nach der Bestätigung der Datenübergabe nach ausgeführter Signaturanfertigung oder Validierung.

Abbildung 4.21 - Screenshot Detailanzeige - Dokument

Alle angezeigten Detailinformationen werden von der Applikation zusammen mit dem Datenblock übertragen. Folgende Detailinformationen werden angezeigt:

- Typ
  Je nach Signaturart können XML-Dokumente nach XML-DSig oder beliebige Datenblöcke nach PKCS#7 signiert werden. Der Typ gibt das Format der übergebenen Datenblöcke nach 6.1.10, DOCUMENT.FORMAT an.
- Meta
  Dieses Feld zeigt die übergebenen Metadaten, wie sie auch in der Statusanzeige angezeigt werden. Der konkrete Inhalt wird von der Applikation vorgegeben und soll ein eindeutiges Ordnungsmerkmal für den Dokumentinhalt tragen.
- SHA256-Hash (Urdokument)
  Hashwert der (exportierten) Urdaten nach SHA256. Dieser Wert kann zur Feststellung der Integrität der (exportierten) Urdaten verwendet werden (siehe 4.3.3, Überprüfung von Dokumenten vor der Signatur).
- SHA256-Hash (Bezugsdokument)
  Hashwert des geladenen Bezugsdokuments (nur bei Verifikation von Dokumenten in der Kodierung PKCS#7, DER, detached)
- SHA256-Hash (Signiertes Dokument)
  Hashwert des signierten Dokuments nach SHA256. Dieser Wert kann zur Feststellung der Integrität des (exportierten) signierten Dokuments herangezogen werden (siehe 4.3.4, Überprüfung von Dokumenten nach der Signatur). Der Führungstext und der Feldinhalt werden nur nach vorangegangener Signatur angezeigt.
- Inhalt
  Freitext zur Anzeige eines weiteren beschreibenden Textes. Inhaltsangabe des Dokuments. Der Inhalt ist von der Applikation vorgegeben.

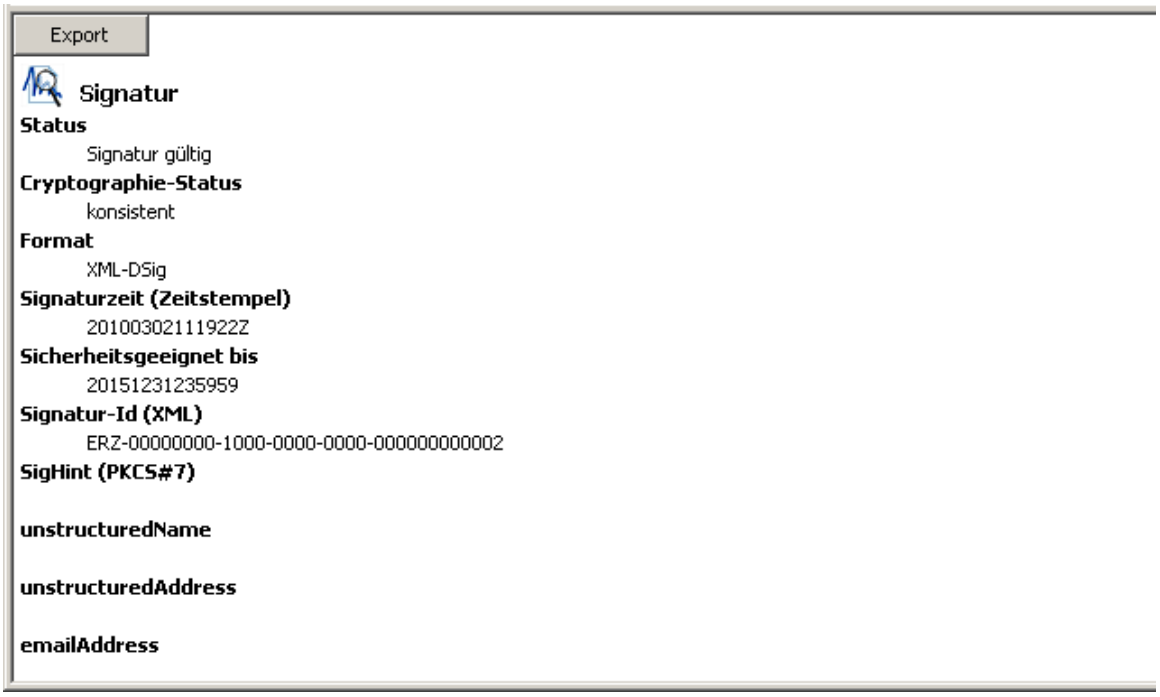
Der Benutzer kann folgende Operationen auf einem Dokument auslösen:

- „Untersuchen“
  Diese Operation bewirkt die Validierung der angezeigten Signaturzertifikate. Zu jedem Zertifikat wird eine OCSP-Auskunft eingeholt, sofern sie nicht bereits vorliegt.
- „Export“
  Mit dieser Operation löst der Benutzer den Export des gesamten Dokuments mit allen abhängigen Daten aus. Vor einer Signaturausführung oder Verifikation liegen lediglich die von der Applikation übergebenen Daten vor. Diese werden zusammen mit einer Beschreibungsdatei nach 6.1.1, XML.CONTENT, stets exportiert. Wenn das Dokument signiert wurde, werden zusätzliche Daten zur ausgeführten Signatur exportiert. Wenn das Dokument verifiziert wurde, werden die Daten zu allen gefundenen Signaturen exportiert. Zum Umfang der exportierten Daten bei Export von Signaturen siehe Folgeabschnitt.

4.2.5.3 Detailanzeige (Signatur)

Die Detailanzeige zur Signatur ist nach ausgeführter Signaturanfertigung oder Verifikation verfügbar. Sie enthält einen Button zur Durchführung der Benutzeraktion “Export“.

Abbildung 4.22 - Screenshot Detailanzeige - Signatur



Die Details der angezeigten Signatur sind in Abhängigkeit vom Signaturformat gefüllt. Stets befüllt sind:

- Status
  Gültigkeit der Signatur nach allen durchgeführten Prüfungen nach 6.1.18, VERIFICATION.STATUS
- Cryptographie-Status
  Gültigkeit der Signatur nach mathematischer Verifikation nach 6.1.12, SIGNATURE.STATUS.
- Format
  Festtest, abhängig vom verwendeten Signaturformat:
  „XML-DSig“ für XML-DSig und „PKCS#7“ für PKCS#7 Signaturen.
- Signaturzeit (<Quelle>)
  Anzeige der Signaturzeit und ihrer Quelle. Als Signaturzeit wird die lokale Systemzeit herangezogen, soweit keine weitere Zeitinformation vorliegt. Wenn die Signatur eine mitsignierte Zeitangabe enthält, wird diese angezeigt. Wenn die Signatur einen Signaturzeitstempel enthält, wird dieser angezeigt. Die Anzeige der Quelle der angezeigten Zeitangabe erfolgt nach 6.1.13, SIG.TIMESOURCE.
- Sicherheitsgeeignet bis
  Anzeige des Ablaufdatums der Sicherheitseignung der verwendeten Signaturmethode. Angezeigt wird das frühste Ablaufdatum der Hashmethode und das Verschlüsselungs-verfahren mit Schlüssellänge.

Optional befüllte Detailangaben

- Id
  Optionales Id-Attribut des Elements Signature, wenn die Signatur im Format XML-dsig vorliegt
- SigHint
  Optionale mitsignierte Signaturrolle bei PKCS#7 Signaturen. Dieses Feld wird befüllt, wenn in der Signatur ein Element „unstructuredName“ nach PKCS#9 vorliegt, jedoch davon abweichend mit ASN.1-Typ OctetString.
- unstructuredName
  Optionales Namenselement nach PKCS#9 (nur bei PKCS#7 Signaturen)
- unstructuredAddress
  Optionales Adressfeld nach PKCS#9 (nur bei PKCS#7 Signaturen)
- emailAddress
  Optionale Mailadresse nach PKCS#9 (nur bei PCKS#7 Signaturen)

Folgende Benutzeraktion ist möglich

- „Export“
  Diese Operation besteht aus dem Export der Signatur (s. 6.1.3, XML.SIGNATURE) und aller abhängigen Detaildaten zu dem verwendeten Signaturzertifikat (s. 6.1.4, XML.CERTIFICATE), einem optionalen Attributzertifikat (s. 6.1.5, XML.ATTRIBCERT), einem optionalen Zeitstempel (s. 6.1.7, XML.TIMESTAMP) und optionalen OCSP Auskünften (s. 6.1.6, XML.OCSP), jeweils mit der Zertifikatskette, soweit diese

bestimmt werden konnte. Zusätzlich werden Detaildaten zu Referenzen exportiert (s. 6.1.3.1, XML.SIGNATURE.XML für XML-DSig und 6.1.3.2, XML.SIGNATURE.PKCS7 für PKCS#7 Signaturen).

4.2.5.4 Detailanzeige (Zertifikat)

Die Detailansicht zu einem (Haupt-) Zertifikat enthält ungeachtet der Herkunft des Zertifikats (von Applikation geladen, in einer Signatur verwendet, Responder- oder Ausstellerzertifikat) stets dieselben Schaltflächen für Benutzeraktionen („OCSP“ und „Export“) und dieselben Detailinformationen.

Abbildung 4.23 - Screenshot Detailanzeige - Zertifikat

Folgende Detailinformationen werden angezeigt:

- Status
  Information zum vorliegenden Prüfergebnis des Zertifikats nach 6.1.17, CERTIFICATE.STATUS. Der Wert „Referenzfehler“ ist für Attributzertifikate vorgesehen und taucht bei Hauptzertifikaten niemals auf.
- OCSP-Status
  Information zum Ergebnis einer OCSP-Prüfung nach 6.1.16, CERTOCSP.STATUS.

- OCSP-Abfragedatum (Systemzeit)
  Systemzeit zum Zeitpunkt der Entgegennahme der OCSP Response nach deren Prüfung.
- Inhaber
  Einzeilige Darstellung der Attribute des Namens des Zertifikatseigentümers aus dem Zertifikat. Der Name kann einige kommaseparierte Attribute enthalten. Attributname und -wert werden durch Gleichheitszeichen „=“ separiert ausgegeben.
- Aussteller
  Einzeilige Darstellung der Attribute des Namens des Zertifikatsausstellers, sonst wie vor.
- Seriennummer
  Eindeutige Seriennummer des vorliegenden Zertifikats beim Aussteller.
- Gültigkeitszeitraum von
  Zeitangabe über den Beginn des Gültigkeitszeitraums des Zertifikats nach dem im Zertifikat hinterlegten Gültigkeitsbeginn.
- Gültigkeitszeitraum bis
  Zeitangabe über das Ende des Gültigkeitszeitraums des Zertifikats nach dem im Zertifikat hinterlegten Gültigkeitsende.
- Attributzertifikate
  Kennzeichnung, ob das Zertifikat die Anbringung von Attributzertifikaten vorsieht „Ja“ oder nicht „Nein“.
- Geeignet für Signaturen
  Kennzeichnung, ob das Zertifikat über die nach Common PKI V2.0 zur Erstellung qualifizierter Signaturen erforderlichen Einträge verfügt „Ja“ oder nicht „Nein“.
- Qualifizierungseintrag

  „Nein“ oder „qcStatement: QcCompliance“ und / oder „Policy: cp-sigconform“. „Nein“ zeigt an, dass das Zertifikat keinen Eintrag enthält, der es als qualifiertes Zertifikat kennzeichnet.
- Selbstbeschränkung
  Attribut „Restriction“ aus dem Zertifikat, Freitext
- Vertretungsmacht
  Attribut „Procuration“ aus dem Zertifikat, in mehrzeiliger Darstellung.
- Weitere Informationen
  Attribut „AdditionalInformation“ aus dem Zertifikat, Freitext
- Zulassung
  Attribut „Admission“ aus dem Zertifikat, in mehrzeiliger Darstellung.

Der Benutzer kann folgende Operationen auf einem Dokument auslösen:

- OCSP
  Diese Operation bewirkt die Einholung einer Online-Statusauskunft zu dem Zertifikat bei

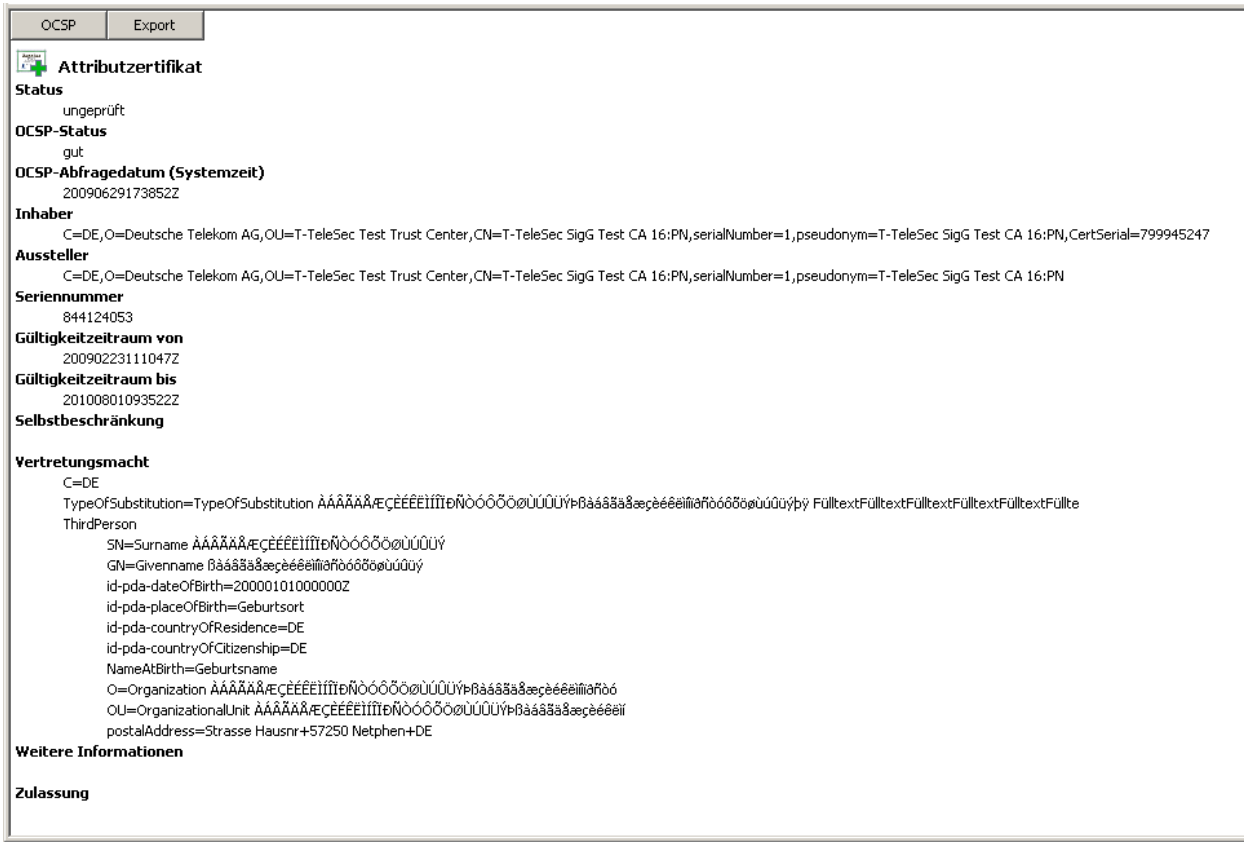
seinem Aussteller. Das Ergebnis der Auskunft wird nach Übernahme in den Feldern „OCSP-Status“ und „OCSP-Abfragedatum“ dargestellt und samt Zertifikatskette in den Navigationsbaum aufgenommen.

- Export
  Durch Auslösen dieser Operation wird das Zertifikat mit seiner Zertifikatskette sowie einer ggf. vorliegenden OCSP Auskunft mit Responderzertifikat und Zertifikatskette exportiert.

4.2.5.5 Detailanzeige (Attributzertifikat)

Die Detailansicht zu einem Attributzertifikat ähnelt der zu einem Zertifikat. Sie enthält dieselben Schaltflächen für Benutzeraktionen

Abbildung 4.24 - Screenshot Detailanzeige - Attributzertifikat



Folgende Detailinformationen werden angezeigt:

- Status
  Information zum vorliegenden Prüfergebnis des Zertifikats nach 6.1.17, CERTIFICATE.STATUS. Der Wert „Referenzfehler“ wird angenommen, wenn sich das Attributzertifikat nicht auf das Signaturzertifikat bezieht.

- OCSP-Status
  Information zum Ergebnis einer OCSP-Prüfung nach 6.1.16, CERTOCSP.STATUS
- OCSP-Abfragedatum (Systemzeit)
  Systemzeit zum Zeitpunkt der Entgegennahme der OCSP-Response nach deren Prüfung.
- Inhaber
  Einzeilige Darstellung der Attribute des Namens des Zertifikatseigentümers aus dem Zertifikat oder Zertifikatsreferenz. Eine Zertifikatsreferenz besteht aus dem Namen des Ausstellers mit der Seriennummer des von ihm ausgestellten Zertifikats, dessen Inhaber auch Inhaber des Attributzertifikats ist. Im Fall der Zertifikatsreferenz ist der Ausstellername bis auf die Zertifikatsnummer mit dem im Folgenden beschriebenen Feld identisch.
  Der Name kann einige kommaseparierte Attribute enthalten. Attributname und -wert werden durch Gleichheitszeichen „=“ separiert ausgegeben.
- Aussteller
  Einzeilige Darstellung der Attribute des Namens des Zertifikatsausstellers.
- Seriennummer
  Eindeutige Seriennummer des vorliegenden Zertifikats beim Aussteller.
- Gültigkeitszeitraum von
  Zeitangabe über den Beginn des Gültigkeitszeitraums des Zertifikats nach dem im Zertifikat hinterlegten Gültigkeitsbeginn.
- Gültigkeitszeitraum bis
  Zeitangabe über das Ende des Gültigkeitszeitraums des Zertifikats nach dem im Zertifikat hinterlegten Gültigkeitsende.
- Selbstbeschränkung
  Attribut „Restriction“ aus dem Attributzertifikat, Freitext
- Vertretungsmacht
  Attribut „Procuration“ aus dem Attributzertifikat, in mehrzeiliger Darstellung.
- Weitere Informationen
  Attribut „AdditionalInformation“ aus dem Attributzertifikat, Freitext
- Zulassung
  Attribut „Admission“ aus dem Attributzertifikat, in mehrzeiliger Darstellung.

Der Benutzer kann folgende Operationen auf einem Dokument auslösen:

- OCSP
  Diese Operation bewirkt die Einholung einer Online-Statusauskunft zu dem Zertifikat bei seinem Aussteller. Das Ergebnis der Auskunft wird nach Übernahme in den Feldern „OCSP-Status“ und „OCSP-Abfragedatum“ dargestellt und samt Zertifikatskette in den Navigationsbaum aufgenommen.
- Export
  Durch Auslösen dieser Operation wird das Zertifikat mit seiner Zertifikatskette sowie

einer ggf. vorliegenden OCSP Auskunft mit Responderzertifikat und Zertifikatskette exportiert.

4.2.5.6 Detailanzeige (Zeitstempel)

Die Detailansicht zu einem Zeitstempel enthält kein Bedienelement für Benutzeraktionen.

Abbildung 4.25 - Screenshot Detailanzeige - Zeitstempel

Ein Zeitstempel dokumentiert das Vorliegen eines Datenblocks zu der in ihm angegebenen amtlichen Zeit. Im Fall eines Signaturzeitstempels wird die Signatur zeitgestempelt, was die späteste Signaturzeit dokumentiert[6].

Folgende Detailinformationen werden angezeigt:

- Status
  Zustand des Zeitstempels nach 6.1.19, TIMESTAMP.STATUS nach allen vorgenommenen Prüfungen. Wenn sich ein Zeitstempel nicht auf den Signaturwert bezieht oder seine Zertifikatskette nicht gebildet werden kann, ist er ungültig.
- Zeit
  signierte (amtliche) Zeitangabe aus dem Zeitstempel
- ImprintHashMethod
  Bei der Hashberechnung des Imprints verwendete Methode
- ImprintHashValue
  Hashwert des Datenblocks

4.2.6 Dateiexporte

Die SAK bietet durch Dateiexporte die Möglichkeit vorgelegte Dokumente vor der Signaturausführung oder bereits signierte und verifizierte Dokumente vor der Übergabe an die Applikation in sämtlichen Details zu prüfen.

[6] Zur Einholung eines Zeitstempels wird dem Zeitstempeldienstanbieter (TSP) ein Imprint (Hashmethode und Hashwert) des zeitzustempelnden Datenblocks vorgelegt, den der TSP signiert zurückgibt.

Der Export wird auf Anweisung des Benutzers durch Betätigen der Schaltfläche “Export“ in einer der Detailanzeigen eingeleitet. Es wird ein Eingabefeld zum Eintrag des Verzeichnisses angezeigt, in den der Export vorgenommen werden soll (hier bereits eingetragen):

Abbildung 4.26 - Screenshot Export - Verzeichniswahl

Durch Betätigung der Schaltfläche “Abbrechen“ wird der Exportvorgang storniert, “Exportieren“ startet den Exportvorgang. Für den Export wird im gewählten Verzeichnis ein Unterverzeichnis mit dem Namen “export_<timestamp>_<lfdnr>“ angelegt, in das die Exportdateien geschrieben werden.

Die beiden variablen Anteile des Unterverzeichnisses sind:

- <timestamp>
  Zeitstempel des Beginns des Exports in der Form “YYYY-MM-DD_HHMISS“ in UTC
- <lfdnr>
  Laufende Nummer des Exports, dreistellig mit führenden Nullen

In dem erstellten Exportverzeichnis findet man nach Fertigstellung des Exports stets die Beschreibungsdatei unter dem Namen “content.xml“ nach 6.1.1, XML.CONTENT und eine Reihe von Binärexporten nach 6.1.23, BINARY.EXPORT. Der Umfang des Exports hängt von dem Objekt ab, auf dessen Detailansicht die Export-Schaltfläche betätigt wurde.

Durch Auslösen des Exports auf der Detailanzeige des Dokuments wird das geladene Urdokument mit allen abhängigen Daten nach 6.1.2, XML.DOCUMENT ausgegeben. Wenn das Dokument gerade signiert wurde, werden zusätzlich das signierte Dokument und die Detaildaten zur angefertigten Signatur ausgegeben (s.u.). Wenn das Dokument verifiziert wurde werden die Detaildaten zu allen ermittelten Signaturen ausgegeben.

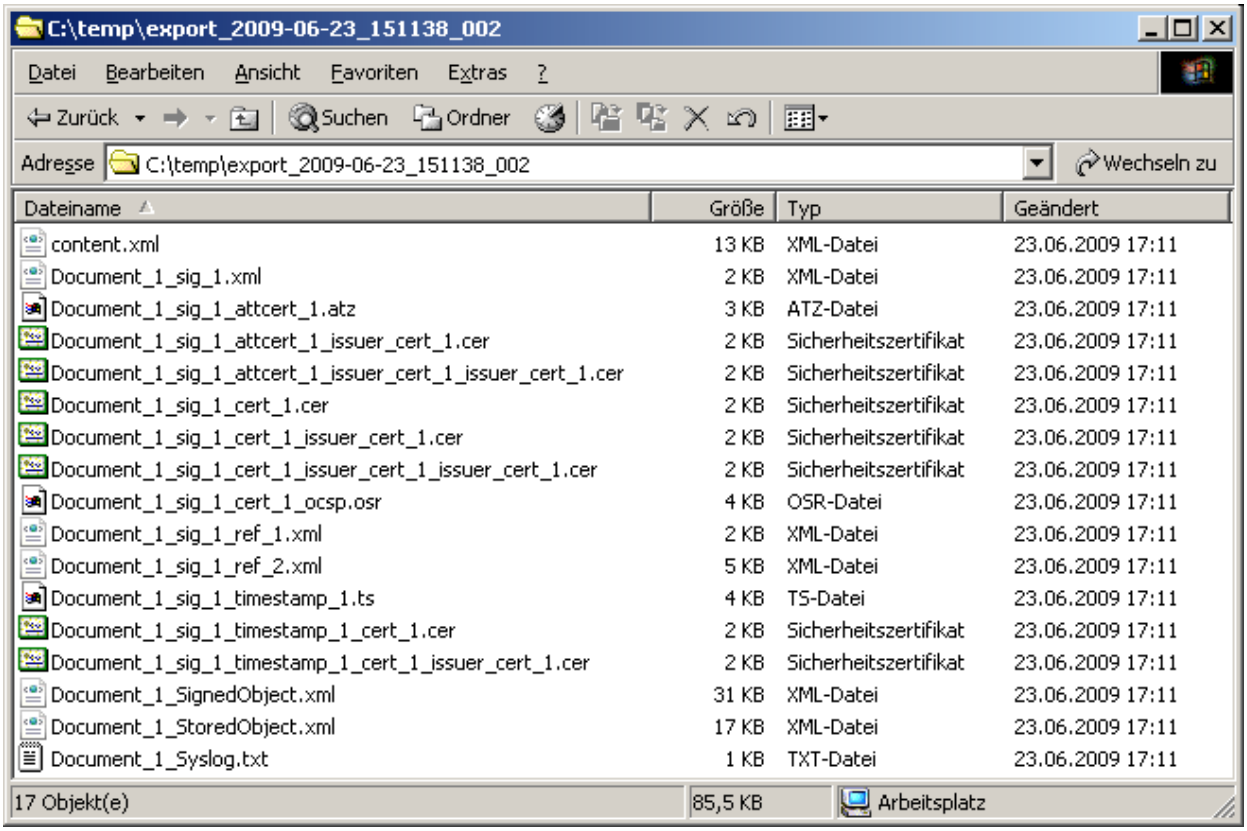
Das Auslösen des Exports auf der Detailanzeige zu einer Signatur bewirkt die Ausgabe der Signaturdaten nach 6.1.3, XML.SIGNATURE mit allen abhängigen Daten, wie Signaturzertifikat, Attributzertifikat und Zeitstempel (s.u.).

Der Export auf der Detailanzeige eines Zertifikats oder Attributzertifikats bewirkt die Ausgabe der Zertifikatsdaten nach 6.1.4, XML.CERTIFICATE bzw. 6.1.5, XML.ATTRIBCERT mit den

Rohdaten des Zertifikats und seiner Zertifikatskette[7] sowie vorliegenden OCSP Responsen, jeweils mit Responderzertifikat und Zertifikatskette.

Im Folgenden ein Beispiel zum Inhalt eines Exportverzeichnisses zu einem mit Signatur- und Attributzertifikat signierten XML Dokument mit Signaturzeitstempel:

Abbildung 4.27 - Screenshot Export - Exportverzeichnisinhalt


C:\temp\export_2009-06-23_151138_002
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Ordner
Adresse C:\temp\export_2009-06-23_151138_002
Wechseln zu
Dateiname Größe Typ Geändert
content.xml 13 KB XML-Datei 23.06.2009 17:11
Document_1_sig_1.xml 2 KB XML-Datei 23.06.2009 17:11
Document_1_sig_1_attcert_1.atz 3 KB ATZ-Datei 23.06.2009 17:11
Document_1_sig_1_attcert_1_issuer_cert_1.cer 2 KB Sicherheitszertifikat 23.06.2009 17:11
Document_1_sig_1_attcert_1_issuer_cert_1_issuer_cert_1.cer 2 KB Sicherheitszertifikat 23.06.2009 17:11
Document_1_sig_1_cert_1.cer 2 KB Sicherheitszertifikat 23.06.2009 17:11
Document_1_sig_1_cert_1_issuer_cert_1.cer 2 KB Sicherheitszertifikat 23.06.2009 17:11
Document_1_sig_1_cert_1_issuer_cert_1_issuer_cert_1.cer 2 KB Sicherheitszertifikat 23.06.2009 17:11
Document_1_sig_1_cert_1_ocsp.osr 4 KB OSR-Datei 23.06.2009 17:11
Document_1_sig_1_ref_1.xml 2 KB XML-Datei 23.06.2009 17:11
Document_1_sig_1_ref_2.xml 5 KB XML-Datei 23.06.2009 17:11
Document_1_sig_1_timestamp_1.ts 4 KB TS-Datei 23.06.2009 17:11
Document_1_sig_1_timestamp_1_cert_1.cer 2 KB Sicherheitszertifikat 23.06.2009 17:11
Document_1_sig_1_timestamp_1_cert_1_issuer_cert_1.cer 2 KB Sicherheitszertifikat 23.06.2009 17:11
Document_1_SignedObject.xml 31 KB XML-Datei 23.06.2009 17:11
Document_1_StoredObject.xml 17 KB XML-Datei 23.06.2009 17:11
Document_1_Syslog.txt 1 KB TXT-Datei 23.06.2009 17:11
17 Objekt(e)
85,5 KB
Arbeitsplatz

Beispiel eines möglichen Inhalts des Exportverzeichnisses nach Export auf der Detailanzeige eines signierten XML Dokuments.

Die Dateinamen sind von der SAK automatisch generiert und mit den in der Beschreibungsdatei “content.xml“ hinterlegten Selektoren identisch. Die Dateinamen werden aus einem Präfix “Dokument_<n>“ (mit <n> = Nummer des Ablagefachs in der SAK, beginnend mit 1), weiteren Namensbestandteilen und einem typabhängigen Suffix gebildet.

Der Suffix von Urdokument und signierten Dokument wird nach dem Dokumentformat (s. 6.1.10, DOCUMENT.FORMAT) unterschieden. Für XML Dokumente ist er fest auf “xml“, für

[7] Jedes Zertifikat ist von seinem Aussteller mit seinem Zertifikat signiert. Daraus ergibt sich eine Kette von Zertifikaten, die durch ein selbstsigniertes Zertifikat (Wurzelzertifikat) abgeschlossen wird. Bei Zertifikatexporten werden stets alle Zertifikate der Kette bis zum (aus der Konfigurationsdatei bekannten) Wurzelzertifikat exportiert.

PKCS# Dokumente fest auf “p7m“ eingestellt. Das geladene Urdokument wird als “Dokument_<n>_StoredObject.<suffix>“ exportiert. Wenn es signiert wurde, wird es unter dem Namen “Dokument_<n>_SignedObject.<suffix>“ exportiert. Beide Dokumente können mit angemessenen Viewern eingesehen werden.

Signaturbezogene Daten tragen als ersten Namensbestandteil stets “Dokument_<n>_sig_<m>“ (mit <m> = laufende Nummer der Signatur, bei XML Dokumenten in der natürlichen Abfolge des Dokuments, bei PKCS#7 Dokumenten von „außen“ nach „innen“, beginnend mit 1).

Zu Signaturen werden fünf unterschiedliche Datenarten nach 6.1.23, BINARY.EXPORT ausgegeben:

- Signaturinhalt
  Selektor: “Dokument_<n>_sig_<m>“ mit Suffix “.xml“ bei XML Dokumenten
  Der in 1. Ordnung signierte Inhalt. Bei XML Dokumenten ist dies der kanonisierte Inhalt des SignedInfo Knotens der Signatur, bei PKCS#7 der Binärkodierte Inhalt der SignedAttributes.
- Signaturreferenz
  Selektor: “Dokument_<n>_sig_<m>_ref_<r>“ mit Suffix “.xml“ bei XML Dokumenten
  Der in 2. Ordnung signierte Inhalt der <r>-ten Referenz (in der Abfolge des Signaturinhalts, beginnend mit 1). Bei XML Dokumenten ist es der Teil des Dokuments, auf den sich die Referenz bezieht, nach Anwendung der im SignedInfo Knoten angegebenen Kanonisierungs- und Transformationsanweisungen. Bei PKCS#7 Dokumenten existiert höchstens eine Signaturreferenz - den signierten Inhalt (das Urdokument).
- Signaturzertifikat
  Selektor: “Dokument_<n>_sig_<m>_cert_<c>.crt“
  Das X509 Zertifikat mit dem die Signatur ausgeführt wurde in Binärform. Die SAK unterstützt keine Co-Signaturen, daher ist <c> stets 1. Exportierte Zertifikate können mit dem Zertifikatsmanager des Betriebssystems in allen Details untersucht werden. Die Zertifikate der zugehörigen Zertifikatskette tragen denselben Namen mit zusätzlich angehängten Bestandteilen “_issuer_cert_1“.
- Attributzertifikat
  Selektor: “Dokument_<n>_sig_<m>_attcert_<a>.atz“
  Attributzertifikat zum Signaturzertifikat in Binärform. Die SAK unterstützt die Verwendung von höchstens einem Attributzertifikat zu einer Signatur, daher ist <a> stets 1. Die Zertifikate der zugehörigen Zertifikatskette tragen denselben Namen mit zusätzlich angehängten Bestandteilen “_issuer_cert_1“.
- Signaturzeitstempel
  Selektor: “Dokument_<n>_sig_<m>_timestamp_<t>.ts“
  Zeitstempel einer Signatur in Binärform. Die SAK unterstützt die Verwendung höchstens eines Signaturzeitstempels zu einer Signatur, daher ist <t> stets 1. Der Imprint des Zeitstempels gehört zu dem Signaturinhalt (s.o.). Die Zertifikate der zugehörigen

Zertifikatskette tragen denselben Namen mit zusätzlich angehängten Bestandteilen "_issuer_cert_1".

Die Datei mit dem Namen "Dokument_<n>_Syslog.txt" enthält textuelle Loginformatinen, vorwiegend zur Fehlerdiagnose im UNIX-Textformat (Zeilenenden ohne <CR>, nur <LF>).

4.2.6.1 Beschreibungsdatei

Bei jedem Export legt die SAK eine Beschreibungsdatei mit festem Namen "content.xml" im Exportverzeichnis an. Bei der Beschreibungsdatei handelt es sich um eine wohlgeformte XML Datei nach 6.1.1, XML.CONTENT ff. mit impliziter Zeichenkodierung nach UTF8. Sie enthält die Namen aller weiteren im Exportverzeichnis angelegten Dateien (in "Selector" Attributen) und stellt sie untereinander in Beziehung.

Sie dient ferner als Informationsquelle wichtiger bei der Ausführung von Signaturen oder deren Verifikation anfallender Informationen. Dazu gehören die Gültigkeit von Signaturen und Zertifikaten, die den Signaturen zugrunde liegenden Funktionen und Parameter sowie Prüfergebnisse.

Die Beschreibungsdatei gibt insbesondere Aufschluss über die in der Statusanzeige wiedergegebenen Verifikations- und Validierungsergebnisse. Im Folgenden sind die Pfade zu den Elementen der Beschreibungsdatei "content.xml" nach 6.1.9, XML.PATH angegeben.

4.2.6.1.1 Verifikationsergebnis und Signaturen

Das in der Statusanzeige dargestellte Verifikationsergebnis zu einem Dokument ist nur dann "gültig", wenn jede untersuchte Signatur "gültig" ist. Wenn das Verifikationsergebnis zu einem Dokument "ungültig" ausweist, findet man anhand der Beschreibungsdatei schnell diejenige Signatur, die zu diesem Ergebnis führt. Die Angaben zu der s-ten Signatur des d-ten Dokuments findet man unter dem Signaturpfad "Content/Document[d]/Signature[s]". Das Ergebnis über alle durchgeführten Verifikationen und Validierungen zu einer Signatur wird im Element <Status> (Signaturstatus) ausgewiesen und kann einen der Werte aus 6.1.12, SIGNATURE.STATUS annehmen.

Der Signaturstatus "Content/Document[d]/Signature[s]/Status" ist "gebrochen", wenn die Nachberechnung des zugrunde liegenden Hashwerts nicht mit dem ausgewiesenen Hashwert übereinstimmt. Wenn dies auf die Signatur 1. Ordnung zutrifft, nimmt das Element "Content/Document[d]/Signature[s]/Consistent" den Wert "false" an.

Der Signaturstatus ist ebenfalls dann "gebrochen", wenn die Nachberechnung eines der Hashwerte der in 2. Ordnung signierten Referenzen nicht mit den ausgewiesenen Hashwerten übereinstimmt. In diesen Fällen nimmt das zugehörige Element

“Content/Document[d]/Signature[s]/Referece[r]/Consistent“ der r-ten Referenz den Wert “false“ an.

Der Wert “ungültig“ wird dem Signaturstatus zugewiesen, wenn das verwendete Signaturzertifikat nicht ermittelt werden konnte, es selbst zum Zeitpunkt der Signatur[8] “Content/Document[d]/Signature[s]/SigningTime“ nach dem anzuwendenden Gültigkeitsmodell ungültig war, nicht zur Anfertigung qualifizierter Signaturen geeignet markiert ist oder seine Zertifikatskette nicht auf ein bekanntes Wurzelzertifikat führt.

Bei XML Signaturen ist ein Signaturstatus “ungesichert“ möglich, der dann angenommen wird, wenn das Signaturzertifikat zwar korrekt verwendet, jedoch nicht mitsigniert wurde. Es werden nur solche Zertifikate als (mögliches) Signaturzertifikat betrachtet, deren Wert im Unterelement <KeyInfo> der Signatur hinterlegt sind. Als mögliche Signaturmechanismen werden alle Referenzen der betrachteten Signatur sowie die XAdES Elemente (unterhalb <Signature> in “.../SigningCertificates/Cert“) berücksichtigt. Im Fall einer Signatur nach XAdES wird der in <CertDigest> ausgewiesene Hashwert über das Zertifikat nachberechnet und muss zu demselben Ergebnis führen. Die in <IssuerSerial> ausgewiesenen Idente sind hinsichtlich der Interoperabilität problematisch. Sie bleiben daher unberücksichtigt,

4.2.6.1.2 Zertifikate und Attributzertifikate

Die SAK berücksichtigt zu jeder Signatur nur ein Signaturzertifikat. Wenn es in der Signatur enthalten ist und ihr zugeordnet werden konnte, ist der Zertifikatsstatus in der Beschreibungsdatei im Unterelement “Certificate/Status“ des Signaturpfades hinterlegt. Die übrigen optional in der Signatur enthaltenen Zertifikate werden nicht berücksichtigt.

Der Gültigkeitszeitraum eines Zertifikats wird bei dessen Verifikation und bei der Einholung von OCSP-Auskünften ggf. korrigiert, sodass die in den Elementen “Certificate/NotBefore“ und “Certificate/NotAfter“ angegebenen Zeiten unmittelbar mit der Signaturzeit verglichen werden können. Das Vergleichsergebnis wird zur Gültigkeitsbewertung des Zertifikats (s.o.) verwendet.

Der Zertifikatsstatus ist “ungültig“, wenn seine Zertifikatskette nicht mit einem bekannten Wurzelzertifikat abschließt oder die Ausstellersignatur für irgendein Zertifikat der Zertifikatskette nicht verifiziert werden kann.

Die Merkmale des Ausstellerzertifikats eines jeden Zertifikats finden sich unter dem Element “Certificate“. Dies gilt für die gesamte Zertifikatskette bis zum selbstsignierten Wurzelzertifikat.

[8] Als Zeitpunkt der Signatur wird der in einem Signaturzeitstempel eingetragene Zeitpunkt verwendet, wenn der Zeitstempel selbst gültig ist und sich auf die Singnatur bezieht. Liegt kein Signaturzeitstempel vor oder ist er nicht gültig, so wird die in der Signatur mitsignierte Zeitangabe verwendet. Wenn keine mitsignierte Zeitangabe vorliegt, wird die lokale Systemzeit als (spätester) Zeitpunkt der Signatur angenommen.

Alle aus der Konfigurationsdatei geladenen Zertifikate erhalten den Zertifikatsstatus “vertrauenswürdig“.

Der Zertifikatsstatus “ungeprüft“ wird als Anfangszustand jedem neu gelesenen Zertifikat zugeordnet. Alle Signaturzertifikate werden geprüft. Sie verbleiben in diesem Zustand nur dann, wenn die mathematische Verifikation des Zertifikats nicht durchgeführt werden kann und tritt im regulären Betrieb nicht auf.

Attributzertifikate enthalten eine Referenz auf das Signaturzertifikat. Attributzertifikate mit beschränkendem Inhalt sollen nach Common-PKI bei Signaturen stets zusammen mit dem Signaturzertifikat verwendet werden. Wenn ein bei einer Signatur verwendetes Attributzertifikat nicht das Signaturzertifikat referenziert, erhält es den Zertifikatsstatus “referenzfehler“. Die SAK hat keine Kenntnis vom Inhalt möglicher Attributzertifikate und ob sie beschränkend wirken. Daher fließt die Gültigkeit eines Attributzertifikats in die Gültigkeitsbewertung einer Signatur nicht ein.

Die Merkmale des Attributzertifikats werden in der Beschreibungsdatei als Element “AttribCert“ unterhalb des Signaturpfades hinterlegt. Auch zu Attributzertifikaten wird die vollständige Zertifikatskette (in dem Unterelement “Certificate“) ausgewiesen.

4.2.6.1.3 Zeitstempel

Zum Nachweis des spätesten Erzeugungszeitpunkts einer Signatur kann zu der Signatur ein Zeitstempel eingeholt werden. Ein Signaturzeitstempel enthält neben der amtlichen Zeit insbesondere einen Hashwert über die Signatur. Die SAK berücksichtigt zu jeder Signatur den ersten vorhandenen Signaturzeitstempel.

Die Merkmale eines Signaturzeitstempels findet man in der Beschreibungsdatei im Im Signaturpfad als Unterelement “TimeStamp“. Der Zeitstempelstatus enthält das vorgenannte Prüfergebnis als Unterelement “Status“. Die SAK fügt der Signatur nur dann einen Signaturzeitstempel hinzu, wenn sie dazu in dem Parametersatz zur Signatur dazu angewiesen wurde und sein Ausstellerzertifikat verifiziert werden konnte. Bei der Signaturprüfung ist die Herkunft eines in die Signaturzeitstempels nicht bekannt, der Zeitstempelstatus wird daher mit “ungeprüft“ vorbelegt.

Bei der nachfolgenden Prüfung erhält er den Zeitstempelstatus “gültig“, wenn die die Nachberechnung des Hashwerts über die Signatur mit dem im Zeitstempel angegebenen Hashwert übereinstimmt und das Ausstellerzertifikat verifiziert werden konnte.

Wenn eine der beiden Prüfungen fehlschlägt, wird dem Zeitstempelstatus “ungültig“ zugewiesen und die enthaltene Zeitangabe bleibt beider Signaturprüfung unberücksichtigt. Das Ausstellerzertifikat wird mit Zertifikatskette im Unterelement “Certificate“ ausgewiesen.

4.2.6.1.4 OCSP Prüfergebnisse

Jedes Zertifikat eines bekannten Ausstellers kann online geprüft werden, sofern dies von der Applikation unterstützt wird. Die Merkmale der Antwort (Response) werden, sofern sie importiert werden konnten, in der Beschreibungsdatei als Unterelement des Zertifikats “OCSP“ eingetragen. Das Unterelement “Status“ von OCSP gibt die Aussage der Response wieder. Wenn das Zertifikat zurückgerufen wurde, sind die Details dazu in den weiteren Unterelementen hinterlegt.

Das Unterelement “OCSPStatus“ des Beschreibungsblocks des Zertifikats nimmt bei vorliegender Response den der Response entsprechenden Wert an. Falls noch keine OCSP Anfrage gestellt wurde, enthält es den Initialwert “ungeprüft“. Wenn der SAK der Aussteller des Zertifikats nicht bekannt ist, wird der Wert “aussteller unbekannt“ eingetragen. Wenn die Verifikation der Response fehlschlägt, enthält es “verifikationsfehler“ und wenn keine interpretierbare Antwort zurückgemeldet wurde den Wert “abfragefehler“.

OCSP Prüfergebnisse können auch zu Attributzertifikaten eingeholt werden. Sie werden analog denen zu den übrigen Zertifikaten bearbeitet.

4.3 Sicherheitshinweise und -empfehlungen

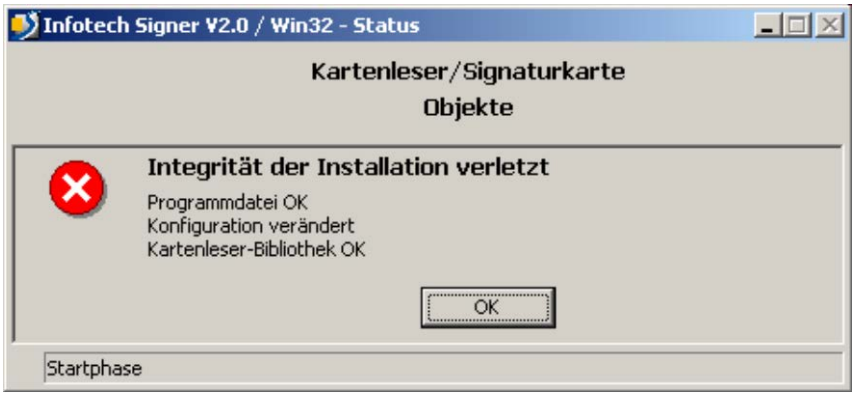
4.3.1 Fehleranzeige und -behandlung

Im Folgenden sind diejenigen Fehlersituationen beschrieben, die im regulären Betrieb nicht auftreten. Zu den Fehlersituationen sind zugehörigen Anzeigen der SAK dargestellt. Unter den Screenshots der Anzeigen sind mögliche Fehlerursachen und Korrekturmaßnahmen beschrieben.

4.3.1.1 Startfehler

Während der Startphase (Reset) erfolgen programminterne Prüfungen. Wenn dabei das Fehlen einer Komponente oder der zugehörigen Signatur festgestellt wird oder die Verifikation der Signatur einer Komponente fehlschlägt, wird dies in der Statusanzeige dargestellt:

Abbildung 4.28 - Screenshot Statusanzeige - Integrität verletzt



Die SAK beendet sich nach Betätigung der Schaltfläche „OK“ sofort, längstens jedoch nach 15 Sekunden selbsttätig.

Wenn diese Fehlersituation eintritt, ist die Integrität der Installation verletzt und kann im Allgemeinen nur durch eine Neuinstallation korrigiert werden. Sofern ausschließlich die Konfigurationsdatei betroffen ist, könnte deren regelmäßig erforderliche Aktualisierung fehlerhaft sein. Dies geschieht insbesondere dann, wenn eine aktive Instanz der SAK jede Änderung der Konfigurationsdatei, nicht jedoch der zugehörigen Signaturdatei, unterbindet.

Es werden die Programmdatei, die Konfigurationsdatei und die Kartentreiber-Bibliothek geprüft und anhand der zugehörigen Signaturdateien verifiziert. Die möglichen Prüfergebnisse lauten:

- “ungeprüft“
  Das Prüfergebnis wurde nicht festgestellt. Dieser Fehler tritt regulär nicht auf. Es besteht der Verdacht der Programmmanipulation!
- “OK“
  Die vorstehende Datei ist authentisch und integer
- “nicht lesbar“
  Die vorstehende Datei wurde nicht gefunden oder kann nicht gelesen werden.
- “Signatur nicht lesbar“
  Die Signaturdatei zu der vorstehenden Datei wurde nicht gefunden oder kann nicht gelesen werden.
- “verändert“
  Authentizität und / oder Integrität der vorstehenden Datei ist verletzt.
- “Schreibschutz versagt“ (nur Konfigurationsdatei)
  Die Einrichtung des Schreibschutzes für diese Datei ist fehlgeschlagen.
- “Datenlesefehler“ (nur Konfigurationsdatei)
  Beim Einlesen der Daten sind Fehler aufgetreten. Die Datei ist unvollständig oder hat das falsche Format.

Falls die Programmdatei oder die Kartenleser-Bibliothek als verändert gemeldet werden, liegt der Verdacht nahe, dass Programmteile durch Schadsoftware verändert wurden. In diesem Fall ist die Ursache der Veränderung dringend zu ermitteln und zu beseitigen.

In der Startphase wird die Aktualität der Konfigurationsdatei geprüft. Ist sie veraltet wird dies auf der Dialogfläche dargestellt:

Abbildung 4.29 - Screenshot Statusanzeige - Konfigurationsdatei veraltet

Die SAK beendet sich nach Betätigung der Schaltfläche „OK“ sofort, längstens jedoch nach 15 Sekunden selbsttätig.

In dieser Fehlersituation muss zunächst die aktuelle Systemzeit geprüft werden. Wenn sie korrekt eingestellt ist, muss anstelle der vorliegenden Konfigurationsdatei eine Aktuelle samt Signaturdatei installiert werden. Die Installation gelingt nur, wenn keine Instanz der SAK die Änderung der Konfigurationsdatei unterbindet. Es sind daher vor der Installation alle laufenden Instanzen der SAK zu beenden. Zudem müssen die Berechtigungen des Anwenders zur Aktualisierung hinreichen. Aktuelle Sätze aus Konfigurationsdatei und zugehöriger Signaturdatei können über die Firmenhomepage des Herstellerunternehmens (über einen sicheren Kanal) geladen werden.

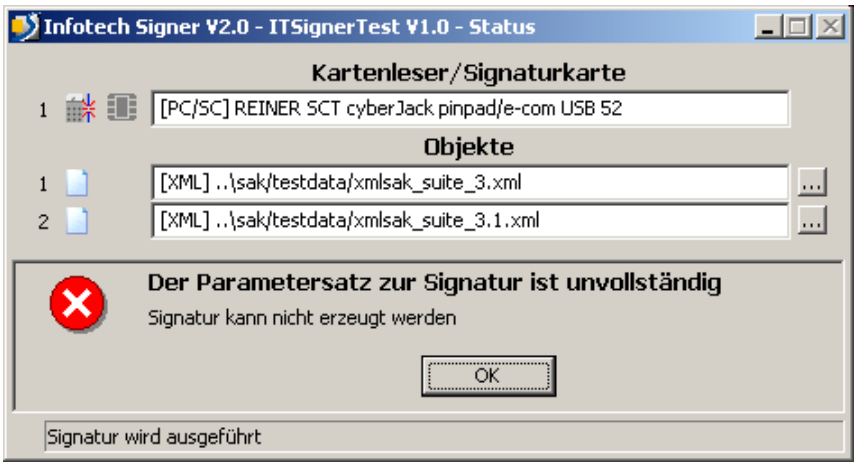
4.3.1.2 Anmeldefehler

Wenn die Anmeldung der Applikation fehlschlägt oder nicht innerhalb einer Minute durchgeführt ist, beendet sich die SAK ohne weitere Anzeige. Ein Benutzereingriff ist nicht möglich.

4.3.1.3 Parameterfehler

Die SAK prüft vor jeder Signatur, ob die von der die SAK verwendenden Fachapplikation übergebenen Signaturparameter zur Signaturausführung hinreichen. Wenn sie nicht alle zur Signatur erforderlichen Angaben enthalten sind, wird dies angezeigt und die Signaturausführung abgebrochen:

Abbildung 4.30 - Screenshot Statusanzeige - Parametersatz unvollständig



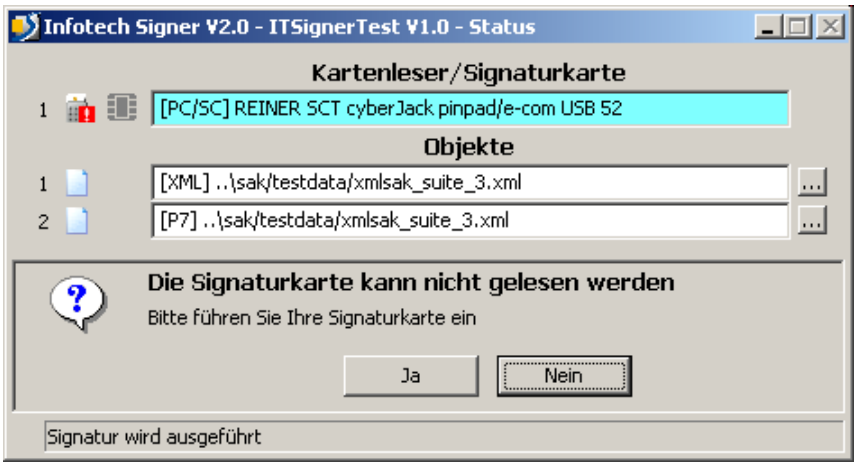
Ein Benutzereingriff ist in dieser Fehlersituation nicht möglich. Die Bestätigung dieses Dialogs leitet die Information über den Abbruch an die aufrufende Applikation weiter.

Es empfiehlt sich, die Aktionen, die zu dieser Fehlermeldung führten, möglicht genau zu dokumentieren und dem Hersteller der Fachapplikation zukommen zu lassen.

4.3.1.4 Signaturkarten

Falls sich in dem Kartenlesegerät keine Signaturkarte befindet, oder die darin befindliche Signaturkarte nicht unterstützt wird, sollte eine andere Signaturkarte verwendet werden und folgender Dialog wird angezeigt:

Abbildung 4.31 - Screenshot Statusanzeige - Signaturkarte nicht gelesen

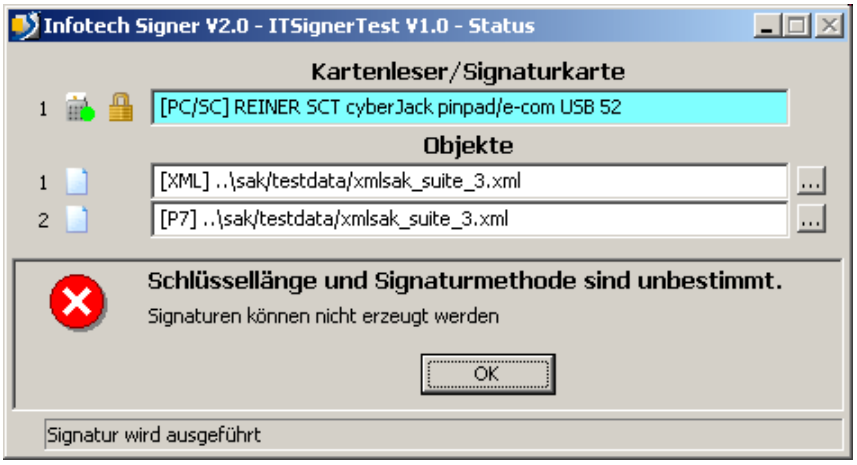


Stellen Sie sicher, dass sich die Signaturkarte in dem Kartenleser befindet und der Kartenleser richtig mit der Arbeitsstation verbunden ist. Prüfen Sie, ob das Kartenlesegerät nicht bereits von einer anderen Applikation belegt ist. Wenn Sie die Signaturkarte aus dem Kartenlesegerät

entfernen und wieder einstecken, wird jede bis dahin bestehende Verbindung zur Signaturkarte getrennt.

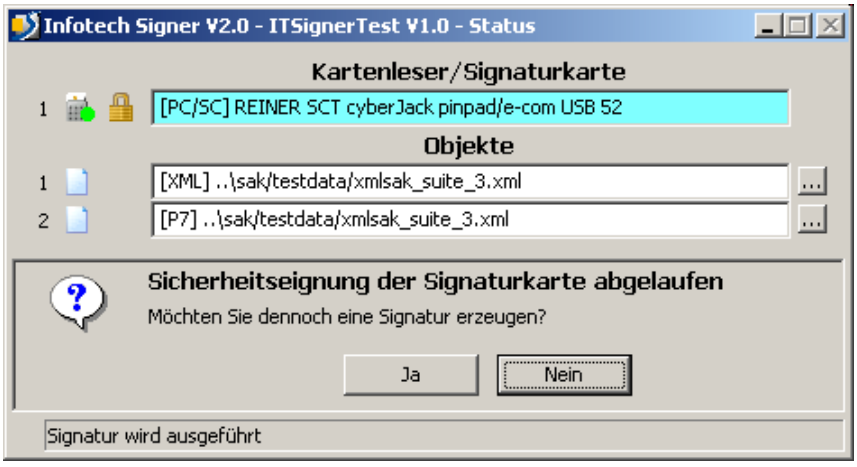
Falls die mit der Signaturkarte ausführbare Signaturmethode nicht ermittelt werden kann oder die Sicherheitseignung der unterstützten Signaturmethode abgelaufen ist wird einer der folgenden Hinweise angezeigt:

Abbildung 4.32 - Screenshot Statusanzeige - Schlüssellänge unbestimmt



In dieser Fehlersituation kann die vorliegende Signaturkarte nicht verwendet werden. Sie ist keine der bekannten Signaturkarten. Verwenden Sie eine der in 2.1, Voraussetzungen aufgelisteten Signaturkarten. Ein Benutzereingriff ist in dieser Fehlersituation nicht möglich. Die Bestätigung dieses Dialogs leitet die Information über den Abbruch an die aufrufende Applikation weiter.

Abbildung 4.33 - Screenshot Statusanzeige - Sicherheitseignung abgelaufen



Diese Fehlersituation tritt ein, wenn Sie eine ältere Signaturkarte verwenden wollen, deren unterstütze Signaturverfahren nicht mehr sicherheitsgeeignet sind. Wenn Sie diesen Dialog

bejahen werden ungültige Signaturen erzeugt, die nicht mehr den aktuellen Sicherheitsstandards entsprechen.

4.3.1.5 Signaturzertifikate

Vor der Signaturausführung wird das zu verwendende Signaturzertifikat aus der Signaturkarte ausgelesen und intern geprüft. Wenn das ausgelesene Signaturzertifikat nicht zur Signatur qualifiziert, nicht vertrauenswürdig oder nicht gültig (noch nicht gültig oder abgelaufen) ist, wird einer der folgenden Hinweise angezeigt:

Abbildung 4.34 - Screenshot Statusanzeige - Zertifikat nicht qualifiziert

Abbildung 4.35 - Screenshot Statusanzeige - Signaturzertifikat nicht vertrauenswürdig

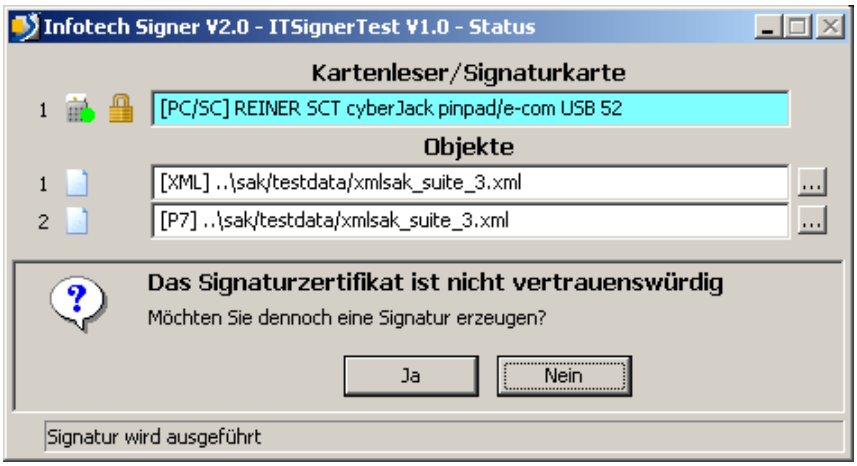


Abbildung 4.36 - Screenshot Statusanzeige - Signaturzertifikat noch nicht gültig

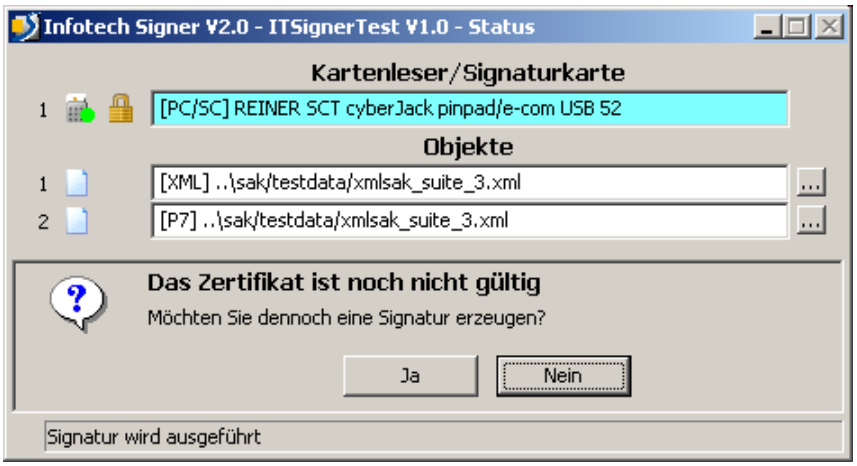
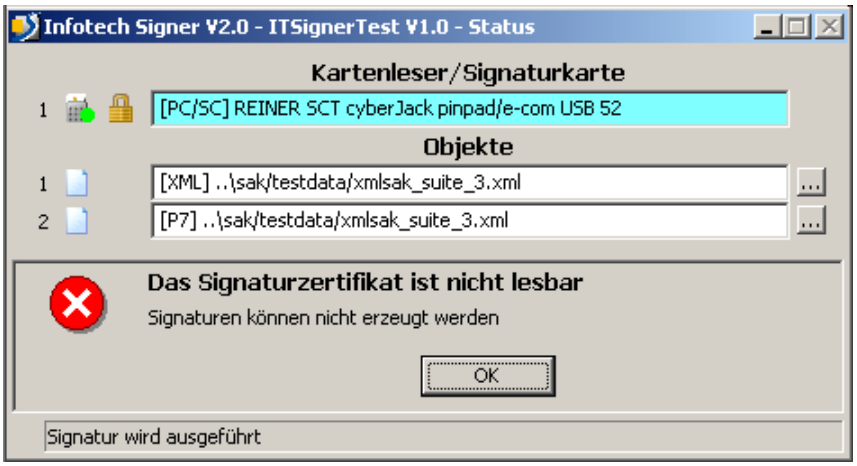


Abbildung 4.37 - Screenshot Statusanzeige - Signaturzertifikat nicht mehr gültig

Falls mehrere der genannten Bedingungen zutreffen, werden die entsprechenden Hinweise nacheinander angezeigt. Wenn einer dieser Dialoge verneint wird, wird der Signaturvorgang abgebrochen. Andernfalls werden ungültige Signaturen erzeugt.

Falls das Signaturzertifikat nicht gelesen werden kann, wird der Signaturvorgang abgebrochen:

Abbildung 4.38 - Screenshot Statusanzeige - Signaturzertifikat nicht lesbar



Ein Benutzereingriff ist in dieser Fehlersituation nicht möglich. Die Bestätigung dieses Dialogs leitet die Information über den Abbruch an die aufrufende Applikation weiter.

4.3.1.6 Attributzertifikate

Bei der Signaturausführung kann dem Signaturzertifikat, wenn es einen entsprechenden Hinweis enthält, auch ein Attributzertifikat hinzugefügt werden. Attributzertifikate werden über die Applikation geladen und vor der Signatur geprüft. Wenn das Attributzertifikat nicht geladen werden konnte, nicht das Signaturzertifikat referenziert oder nicht gültig (noch nicht gültig oder abgelaufen) ist, wird einer der folgenden Hinweise angezeigt:

Abbildung 4.39 - Screenshot Statusanzeige - Attributzertifikat Ladefehler

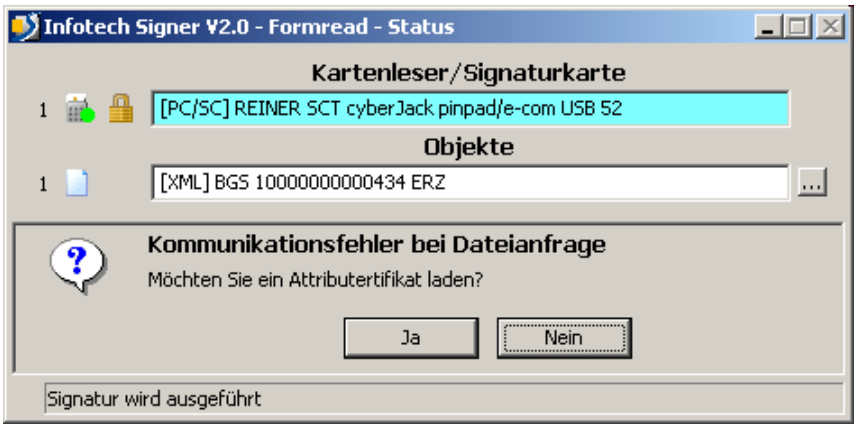


Abbildung 4.40 - Screenshot Statusanzeige - Attributzertifikat Referenzfehler

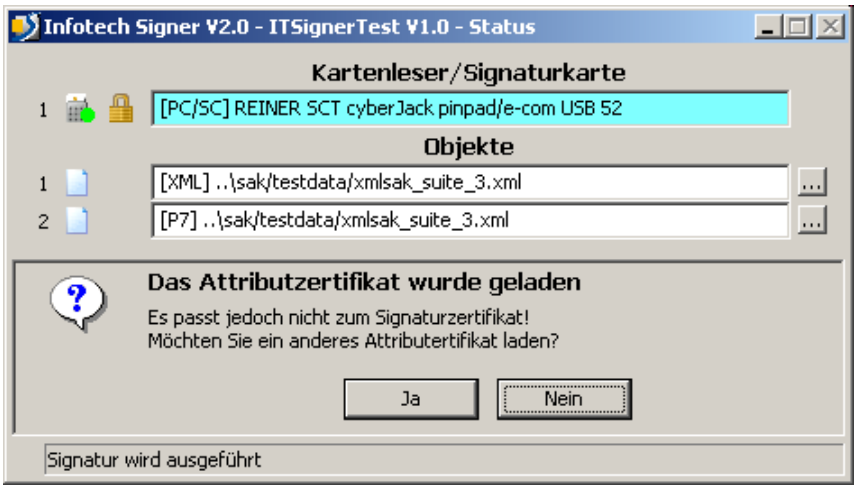


Abbildung 4.41 - Screenshot Statusanzeige - Attributzertifikat noch nicht gültig

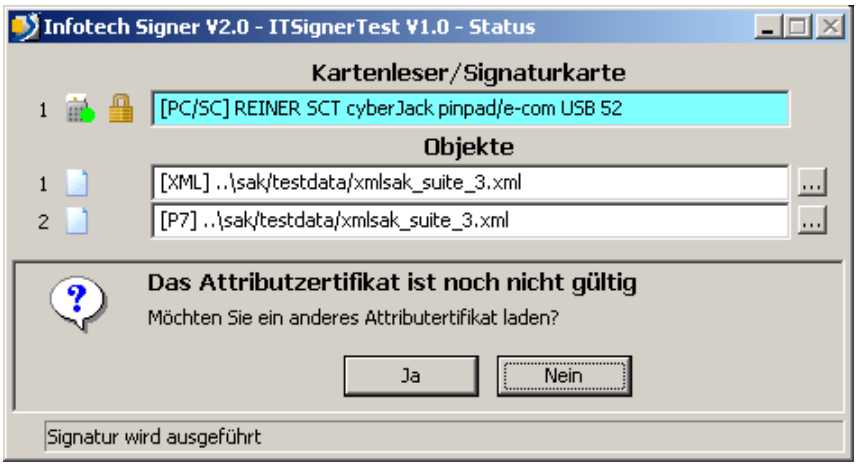
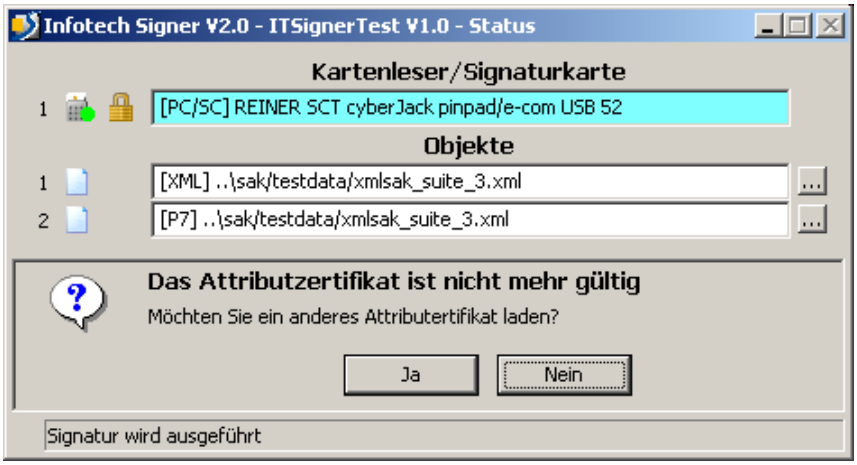


Abbildung 4.42 - Screenshot Statusanzeige - Attributzertifikat nicht mehr gültig

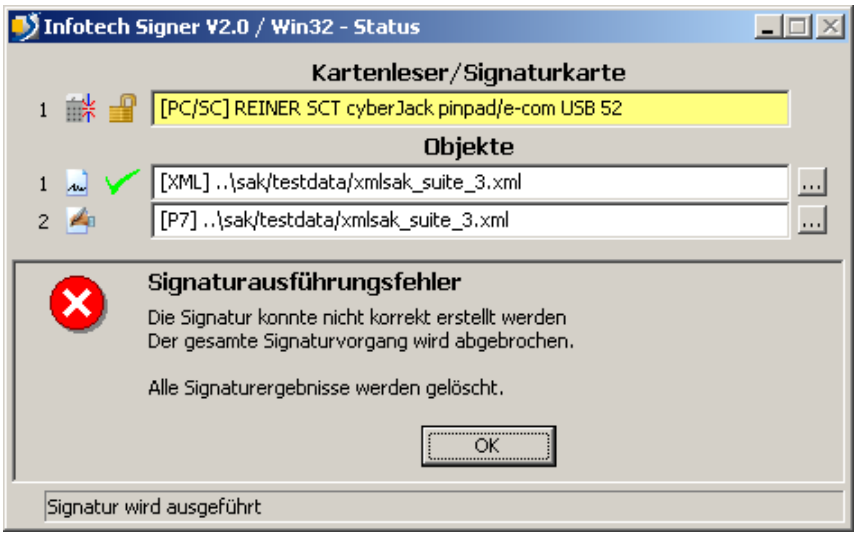


Bei Bejahung eines der dargestellten Dialoge wird der Ladevorgang wiederholt. Bei Verneinung wird der Signaturvorgang ohne Attributzertifikat fortgesetzt.

4.3.2 Automatische Prüfung von Dokumenten nach der Signatur

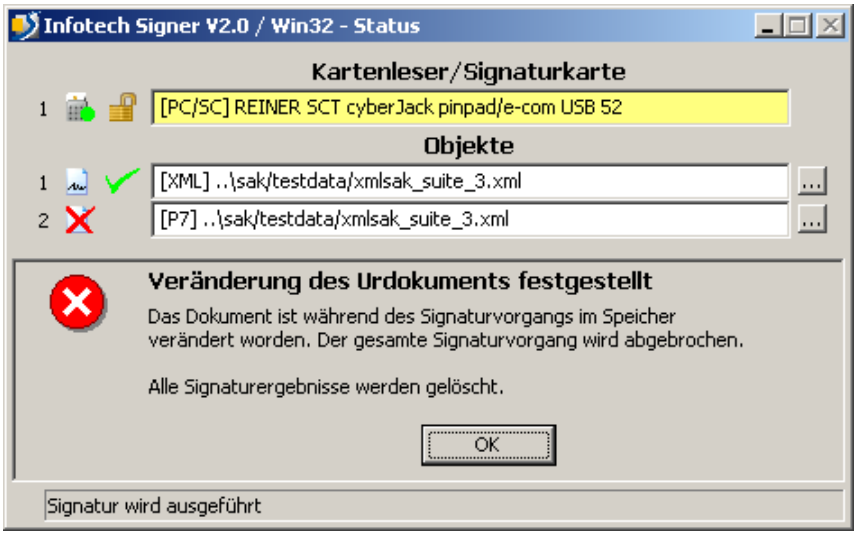
Unmittelbar nach der Signaturausführung erfolgt eine automatische Prüfung, ob die angebrachte Signatur korrekt ist. Wenn die Signatur nicht korrekt ausgeführt werden konnte, wird der folgende Dialog angezeigt und alle bis dahin erzeugten Signaturergebnisse verworfen.

Abbildung 4.43 - Screenshot Statusanzeige - Signaturausführungsfehler



Es wird ferner geprüft, ob das geladene Urdokument bis zum Abschluss des Signaturvorgangs unverändert geblieben ist. Falls eine Veränderung festgestellt wird, wird der nachfolgende Dialog angezeigt und alle bis dahin erzeugten Signaturergebnisse verworfen.

Abbildung 4.44 - Screenshot Statusanzeige - Urdokument verändert



4.3.3 Überprüfung von Dokumenten vor der Signatur

Um ein geladenes Dokument vor der Signaturausführung einzusehen, bricht man die PIN-Eingabe ab (z.B. durch Betätigung des “Abbruch“-Knopfes auf dem Tastenfeld des Kartenlesegeräts), fordert mit dem Navigationselement (…) der Statusanzeige (siehe 4.2.3.1, Statusanzeige bei Signaturausführung) die Detailanzeige des interessierenden Dokuments an und führt dessen Export durch (siehe 4.2.6, Dateiexporte). Im Exportverzeichnis findet man unter dem Namen “Dokument_<n>_StoredObject“ (mit Suffix “.xml“ oder “.p7m“) das zur Signatur anstehende Dokument.

Die Integrität des (exportierten) Urdokuments kann anhand des in der Detailanzeige ausgewiesenen SHA-256 Hashwerts geprüft werden (siehe 4.2.5.2, Detailanzeige (Dokument)).

4.3.4 Überprüfung von Dokumenten nach der Signatur

Nachdem eine Signatur ausgeführt wurde, muss die Übertragung der Ergebnisdaten an die Applikation bestätigt werden (siehe 4.2.3.1, Statusanzeige bei Signaturausführung). Während der Anzeige des Bestätigungsdialogs führt man den Export des Dokuments durch, wie im vorangegangenen Abschnitt beschrieben. Nach durchgeführtem Export enthält das Exportverzeichnis u.a. das signierte Dokument unter dem Namen “Dokument_<n>_SignedObject“ (mit Suffix “.xml“ oder “.p7m“) und eine Zusammenstellung signaturrelevanter Datenextrakte (siehe 4.2.6, Dateiexporte ff.).

Die Integrität des (exportierten) signierten Dokuments kann anhand des in der Detailanzeige des Dokuments ausgewiesenen SHA-256 Hashwerts geprüft werden (siehe 4.2.5.2, Detailanzeige (Dokument)).

Der signierte Anteil eines gerade signierten Dokuments wird idR. durch die erste Referenz unter dem Namen “Dokument_<n>_sig_1_ref_1“ (mit Suffix “.xml“ oder “.p7m“) exportiert. Der Hashwert der referenzierten Daten wird in der Beschreibungsdatei zusammen mit dem Hashverfahren ausgewiesen.

Die SAK hat keine Kenntnis über das Schema (im Fall eines signierten XML Dokuments) oder das Datenformat (bei PKCS#7 Signaturen). Der Datengehalt des von der Signatur überdeckten Anteils wird stets als Ganzes exportiert. Der Bedeutungsgehalt dieser Daten ist nicht Gegenstand der SAK.

5 Benutzerhandbuch für den Entwickler

Für Entwickler gelten die Ausführungen aus Kapitel 4, dem Benutzerhandbuch für den Endanwender. Die folgenden Abschnitte dieses Kapitels enthalten die für den Entwickler relevanten Ergänzungen.

5.1 Sicherheitsfunktionen

Die Beschreibung der Sicherheitsfunktionen ist mit derjenigen aus 4.1, Sicherheitsfunktionen identisch. Der vorgenannte Abschnitt ist gleichfalls Bestandteil dieses Handbuchs.

5.2 Benutzerschnittstelle

Die Beschreibung der Oberflächenteile muss hinsichtlich der Namensgebung der SAK und des Programmicons gegen die bei der Registrierung der Fachapplikation mit dem Hersteller vereinbarten ausgetauscht werden. Mit der Veröffentlichung ist das aktualisierte Benutzerhandbuch verfügbar zu machen.

Der Austausch des Programmicons und des Programmnamens hat keinerlei Auswirkungen auf die Sicherheitsfunktionalitäten der SAK.

Alle für den Endbenutzer verfügbaren Informationen sind auch von der die SAK verwendenden (Fach-) Applikation durch eine Schnittstelle abrufbar. Die Schnittstellenbeschreibung befindet sich im Anhang 6.2, Applikationsschnittstelle und ist Bestandteil dieses Handbuchs.

5.3 Sicherheitshinweise und -empfehlungen

5.3.1 Fehlermeldung und -behandlung

5.3.1.1 Startfehler

Während der Startphase (Reset) erfolgen programminterne Prüfungen. Wenn dabei das Fehlen einer Komponente oder der zugehörigen Signatur festgestellt wird, die Verifikation der Signatur einer Komponente fehlschlägt oder die Konfigurationsdatei veraltet ist, beendet sich die SAK auf Benutzeranforderung, sonst längstens nach 15 Sekunden selbstständig.

Während dieser Zeit zeigt die SAK dem Benutzer in der Statusanzeige den Grund des Abbruchs (siehe 4.3.1.1, Startfehler).

Unmittelbar vor dem Abbau der Applikationsschnittstelle (siehe 6.2 und 6.2.4, Aktivierungsphase) setzt sie anstelle der Bereitschaftsmeldung eine Fehlermeldung ab.

Die Fehlermeldung lautet: “Integrität der Installation verletzt“, wenn die Integrität der Installation nicht verifiziert werden konnte, oder "Konfigurationsdatei veraltet", wenn die Konfigurationsdatei veraltet ist.

Die Applikation sollte den Benutzer mindestens darauf hinweisen, dass die Signatur- und Verifikationsfunktionen nicht zur Verfügung stehen, besser jedoch über die kommunizierte Abbruchursache Auskunft erteilen.

Wenn die Konfigurationsdatei veraltet ist, sollte die Applikation die Aktualisierung Konfigurationsdatei und der zugehörigen Signaturdatei unterstützen (ggf. durch Download aus der in 2.7, Wartung und Pflege angegebenen Quelle) und die SAK erneut starten.

5.3.1.2 Anmeldefehler

Sobald die SAK sie Selbstprüfung und in Inventarisierung der Kartenlesegeräte abgeschlossen hat, leitet sie die Anmeldephase ein. Die Anmeldung erfolgt durch das Kommando “Activate“ (siehe 6.2.5, Funktionen der SAK) von der Applikation zur SAK.

Wenn die Applikation das Kommando “Activate“ nicht innerhalb von 5 Sekunden absetzt, terminiert die SAK mit der Fehlermeldung “Authentisierung überfällig“. Andernfalls prüft die SAK die mit dem Kommando übergebene Authentisierung (siehe 6.2.4, Aktivierungsphase). Wenn die Prüfung fehlschlägt, terminiert die SAK mit der Fehlermeldung “Authentisierung fehlgeschlagen“.

Die hier beschriebenen Fehler sollten im regulären Betrieb nicht auftauchen. Falls sie dennoch vorkommen, setzen Sie sich zur Registrierung Ihrer Applikation mit der Herstellerfirma in Verbindung.

5.3.1.3 Parameterfehler

Der mit dem Kommando StoreObject übergebene Parametersatz nach 6.1.8, XML.STOREPARAM wird insbesondere zum Transport der Signaturparameter verwendet. Wenn die übergebenen Daten verifiziert werden sollen, darf der Parametersatz leer sein, sollte jedoch stets die Inhaltsangabe (Element <Description>) zu den übergebenen Daten enthalten.

Zu Signaturzwecken übergebe Parametersätze müssen wohlgeformt sein und im Fall von XML Signaturen genau ein Element <XmlSignature>, im Fall vom PKCS#7 Signaturen genau ein

Element <PKCS7Signature> enthalten. Über die Art der Signatur entscheidet das mit StoreObject übergebene Datenformat.

5.3.2 Wahl geeigneter Metadaten

Für den Endbenutzer ist es von entscheidender Bedeutung den Inhalt zu signierender Dokumente anhand der Metadaten, die mit dem Dokument zusammen übertragen werden, eindeutig identifizieren zu können.

Die Metadaten müssen daher für jedes vorgelegte Dokument eindeutig, mit möglichst engem Bezug zu dem Inhalt gewählt werden. Mögliche Idente sind z.B. Rechnungsnummer, Tagesdatum, Dokumentart, Vorgangsnummer, etc..

Zusammen mit den Dokument- und Metadaten kann die Applikation im Parameterblock eine Inhaltsangabe mitteilen (siehe 6.1.8, XML.STOREPARAM, Element <Description>), die in der Detailanzeige des Dokuments mit dem Führungstext „Inhalt“ ausgegeben wird. Die Inhaltsangabe ist zum Transport weiterer Detailinformationen über das Dokument vorgesehen.

6 Anhang

6.1 Datendefinitionen

Der Applikation wird über E.APP der gleiche Umfang an Daten angeboten wie dem Benutzer über E.DISPLAY und die Exportfunktionen. Die verwendeten Datenstrukturen werden hier übergreifend beschrieben.

In den Abschnitten 6.1.1 bis 6.1.8 werden die XML Strukturen (Container) beschrieben, wie sie der Applikation über E.APP oder über E.FILE als Datei bzw. Datenblock bereitgestellt werden. Abschnitt 6.1.9 beschreibt eine Konvention zur Angabe einer Position innerhalb eines XML Dokuments. In den restlichen Abschnitten 6.1.10 bis 6.1.25 sind die Aufzählungstypen mit Kurzbezeichnung beschrieben.

In der Strukturbeschreibung werden die folgenden grundlegenden Datenformate verwendet:

Tabelle 6.1 - Einfache Datenformate

Formatbezeichnung	Bedeutung
Bool	Wahrheitswert “true” oder “false”
DateTime	Datumsangabe in “YYYYMMDDHHMISS”, optional mit angehängtem “Z“ zur Markierung der Zeitangabe als UTC
HexString	String aus [“0“..“9“], [“a“..“f“]
Int	String aus optionalem Vorzeichen (“+“|“-“) und Dezimalzeichen [“0“..“9“]
String	Kette beliebiger UTF8-kodierter Unicode Zeichen (RFC3629)

6.1.1 XML.CONTENT

Diese Struktur umfasst den vollständigen Umfang der in der SAK gespeicherten Informationen. Als Unterelemente stehen alternativ das Dokument und das Zertifikat zur Verfügung

<Content TimeOfExport>

- <Document> XML.DOCUMENT
- <Certificate> XML.CERTIFICATE

Content	Container, Wurzelelement der Übersicht
TimeOfExport	DateTime, Systemzeit zum Zeitpunkt des Exports
Document	Container, optionales Unterelement für ein einzelnes abgelegtes Dokument und seine Begleitdaten nach XML.DOCUMENT
Certificate	Container, optionales Unterelement für ein einzelnes abgelegtes Zertifikat und seine Begleitdaten nach XML.CERTIFICATE.

6.1.2 XML.DOCUMENT

Diese Struktur beinhaltet die Übersichtsdaten zu einem geladenen Dokument.

<Document Meta Format>

- <Signature> XML.SIGNATURE
- <SignedObject Selector>
- <StoredObject Selector>
- <StoredSubdata Selector>
- <Status Info> DOCUMENT.STATUS
- <Log Selector>

Document	Container, Wurzelelement der Übersichtsdaten zu einem Dokument
Meta	String, die bereitgestellten Meta-Daten
Format	Erkanntes Datenformat nach DOCUMENT.FORMAT
Signature	Container, Informationen zu Signaturen nach XML.SIGNATURE
SignedObject	stellt den Selektor für das signierte Objekt/Dokument bereit
StoredObject	stellt den Selektor für das ursprünglich geladene Objekt bereit
StoredSubdata	stellt den Selektor für das Bezugsdokument bei detached Signaturen
Status	Gesamtstatus des Dokuments nach DOCUMENT.STATUS
Info	String, textuelle Darstellung des Status (s.v.)
Log	stellt den Selektor für das Systemlog zum Dokument
Selector	Selektor für den zugehörigen Datenblock

6.1.3 XML.SIGNATURE

Übersichtsdaten zu einer Signatur.

<Signature Selector>

<Status Info> VERIFICATION.STATUS
<CryptoStatus Info> SIGNATURE.STATUS
<SigningTime > DateTime
<SigningTimeSource Info> SIG.TIMESOURCE
<SignatureMethod KeyLength> String
<Consistent> Bool
<SecureUntil> DateTime
<SignedHash> HexString
<SignedHashMethod> HASH.METHOD
<SignedHashValue> HexString
<SignedHash> HexString
<XMLSignature ID Path> XML.SIGNATURE.XML
<PKCS7Signature Mime Detached Selector> XML.SIGNATURE.PKCS7
<Certificate> XML.CERTIFICATE
<AttribCert> XML.ATTRIBCERT
<TimeStamp> XML.TIMESTAMP

Signature	Container, Wurzelelement der Signaturübersicht
Selector	Selektor für den Datenblock, auf den sich der Signaturwert bezieht (Signatur 1. Ordnung). Abhängig vom Signaturtyp: SignedInfo-Knoten, bei XML, Signed Attributes (oder Urdokument) bei PKCS#7.
Status	Ergebnis der Signaturprüfung nach VERIFICATION.STATUS
Info	String, textuelle Darstellung des Status (s.v.)
CryptoStatus	Ergebnis der Signaturprüfung nach SIGNATURE.STATUS
Info	textuelle Darstellung des CrypoStatus (s.v.)
SigningTime	DateTime, Signaturzeit aus Zeitstempel (sofern vorhanden und gültig), sonst aus Eintragung in der Signatur (i.d.R Systemzeit, sofern vorhanden), sonst aktuelle Systemzeit.
SigningTimeSource	Angabe der Quelle der Signaturzeit (s.v.) nach SIG.TIMESOURCE
Info	String, textuelle Darstellung der Quelle der Signaturzeit (s.v.)
SignatureMethod	Algorithmen und Parameter, die zur Signaturerzeugung genutzt wurden s. SIGNATURE.METHOD.
KeyLength	Länge des Signaturschlüssels des verwendeten Cryptographischen Verfahrens. Hier: RSA PKCS#1, Schlüssellänge = Länge Modulus
Consistent	Aussage, ob die Signatur 1. Ordnung integer (ungebrochen) ist. Wenn true, reproduziert die Nachberechnung des Hashwerts des Signaturgegenstands 1. Ordnung (s.o. Selector) den im SignedHash enthaltenen und mit SignedHashValue ausgewiesenen Hashwert

SecureUnltil	Frühester Zeitpunkt, an dem die Sicherheitseignung der Hashmethode oder des Verschlüsselungsverfahrens ausläuft.
SignedHashMethod	Algorithmus zur Hashwertbildung nach HASH.METHOD aus dem entschlüsselten Signaturwert (s.u.: SignedHash).
SignedHashValue	Hex-Darstellung des Hashwerts aus dem entschlüsselten Signaturwert (s.u.: SignedHash)
SignedHash	Hex-Darstellung des Signaturwerts
XMLSignature	Container, nach XML.SIGNATURE.XML (nur bei XML-Signaturen)
PKCS7Signature	Container, nach XML.SIGNATURE.PKCS7 (nur bei PKCS#7-Signaturen)
Certificate	Container, Daten des ersten Signaturzertifikats, siehe XML.CERTIFICATE
AttribCert	Container, Daten des ersten Attributzertifikats, siehe XML.ATTRIBCERT
TimeStamp	Container, Daten des ersten Zeitstempels, siehe XML.TIMESTAMP

6.1.3.1 XML.SIGNATURE.XML

Struktur spezifischer Übersichtsdaten zu XML-Signaturen. Die Referenzknoten entstammen dem SignedInfo Element. Sie sind von der Signatur umschlossen (Signatur 2. Ordnung).

<XMLSignature ID Path>

<Reference URI Type XAdES Selector>

<Consistent> Bool
<SecureUntil> DateTime
<HashMethod> HASH.METHOD
<HashValue> HexValue
<CoveredSignatures> String

XMLSignature	Container, Wurzelement XML-spezifischer Signaturdaten
ID	String, die ID aus dem Signature-Knoten, sofern vorhanden
Path	String, absoluter Pfad zum Signature-Knoten nach XML.PATH.
Reference	Container, Wurzelelement eines Reference-Knotens
URI	String, im Reference-Knoten eingetragenes URI
Type	String, im Reference-Knoten verwendetes Type-Attribut
XAdES	Kennzeichnung true, wenn der Reference-Knoten nach XAdES angelegt wurde, andernfalls false
Selector	Selektor für die referenzierten Daten (Signatur 2. Ordnung).
Consistent	Aussage, ob die Signatur 2. Ordnung integer (ungebrochen) ist Wenn true, reproduziert die Nachberechnung des Hashwerts nach der in HashMethod ausgewiesenen Methode auf den referenzierten Daten

	(s.o. Selector) den in der Referenz enthaltenen und mit HashValue ausgewiesenen Wert
SecureUntil	Zeitpunkt, an dem die Sicherheitseignung der Hashmethode ausläuft.
HashMethod	Algorithmus zur Hashwertbildung s. HASH.METHOD
HashValue	HEX-Darstellung des im Dokument gefundenen Hashwertes
CoveredSignatures	Leerzeichen-separierte Liste der mit dieser Signatur umschlossenen Signaturen innerhalb des Dokuments nach Reihenfolge des Auftauchens im Dokument, beginnend mit 1

6.1.3.2 XML.SIGNATURE.PKCS7

Struktur spezifischer Übersichtsdaten zu PKCS#7-Signaturen.

<PKCS7Signature Mime Detached Selector>

<Reference Selector>

<Consistent> Bool
<SecureUntil> DateTime
<HashMethod> HASH.METHOD
<HashValue> HexString

<sigHint> String
<unstructuredName> String
<unstructuredAddress> String
<emailAddress> String

PKCS7Signature	Container, Wurzelelement PKCS#7-spezifischer Signaturdaten
Mime	Bool, true, wenn ein Ausgabeformat nach S/MIME vorliegt
Detached	Bool, true, wenn die Signatur als detached signature vorliegt
Reference	Container für die signierten Urdaten (Signatur 2. Ordnung)
Selector	Selektor für den Datenblock (das Urdokument)
Consistent	Aussage, ob die Signatur 2. Ordnung integer (ungebrochen) ist. Wenn true, reproduziert die Nachberechnung des Hashwerts nach der in HashMethod ausgewiesenen Methode auf den referenzierten Daten (das Urdokument, s.o. Selector) den in der Referenz enthaltenen und mit HashValue ausgewiesenen Wert
SecureUntil	Zeitpunkt, an dem die Sicherheitseignung der Hashmethode ausläuft.
HashMethod	Algorithmus zur Hashwertbildung s. HASH.METHOD
HashValue	HEX-Darstellung des im Dokument gefundenen Hashwertes
SigHint	String, wird aus dem ersten String von unstructuredName nach PKCS#9 belegt, sofern vorhanden und der ASN1-Typ gleich OCTET_STRING ist.

unstructuredName	String, wird aus dem ersten String von unstructeredName nach PKCS#9 belegt, sofern vorhanden.
unstructuredAddress	String, wird aus dem ersten String von unstructuredAddress nach PKCS#9 übernommen, sofern vorhanden.
emailAddress	String, wird aus dem ersten String von emailAddress nach PKCS#9 übernommen, sofern vorhanden.

6.1.4 XML.CERTIFICATE

Diese Struktur beschreibt jeweils ein Zertifikat und seine Begleitdaten

<Certificate Selector>

<Status Info> Int
<OCSPStatus Info Sysdate> Int
<Subject> String
<NotBefore> DateTime
<NotAfter> DateTime
<Serial> String
<Issuer> String
<Certificate> XML.CERTIFICATE
<AttributeCertificate> Bool
<QualifiedForSignature> Bool
<QualificationEntries> String
<Procuration> String
<Restriction> String
<AdditionalInformation> String
<Admission> String
<OCSP> XML.OCSP

Certificate	Container, Beschreibungsblock für ein Zertifikat.
Selector	Ein Ident für das Zertifikat, über das die Binärform des Zertifikats über E.APP angefordert werden kann. Beim Export bestimmt dieses den angehängten Anteil des Dateinamen.
Status	Ergebnis der Zertifikatsprüfung nach CERTIFICATE.STATUS
Info	textuelle Darstellung des Status (s.v.)
OCSPStatus	Ergebnis der Online Zertifikatsprüfung nach CERTOCSP.STATUS
Info	textuelle Darstellung des OCSPStatus (s.v.)
Sysdate	Systemdatum zur Einholung der OCSP-Information
Subject	voll qualifizierter Name des Karteninhabers

NotBefore	Beginn des Gültigkeitszeitraums[9] in UTC
NotAfter	Ende des Gültigkeitszeitraums[10] in UTC
Serial	Seriennummer des Zertifikats
Issuer	voll qualifizierter Name des Ausstellers
Certificate	Container, Ausstellerzertifikat nach XML.CERTIFICATE (sofern verfügbar.
AttributeCertificates	Wert von LiabilityLimitationFlag, default „false“
QualifiiedForSignature	true, wenn das Zertifikat für die Erstellung von Signaturen geeignet ist.
QualificationEntries	“`qcStatement: QcCompliance`“ und / oder “`Policy: cp-sigconform`“, sofern das Zertifikat über die entsprechenden Eintragungen verfügt.
Procuration	Angaben zur Vertretungsmacht
Restriction	Selbstbeschränkung, Freitext
AdditionalInformation	Zusatzinformationen, Freitext
Admission	Angaben zur beruflichen Zulassung
OCSP	Container, Daten eines OCSP-Checks nach XML.OCSP, wenn eine OCSP-Check erfolgreich durchgeführt wurde

6.1.5 XML.ATTRIBCERT

Diese Struktur beschreibt jeweils ein Zertifikat und seine Begleitdaten

[9] Der hier ausgewiesene Gültigkeitszeitraum ist der im Zertifikat ausgewiesene Zeitraum, jedoch um die ggf. durchgeführte Verifikation und ein ggf. vorliegenden OCSP Prüfergebnis korrigiert. Aus dem Ausstellerzertifikat wird der Gültigkeitszeitraum nach dem in ihm ausgewiesenen Gültigkeitsmodell reduziert: beim Schalenmodell auf den Zeitraum, in dem beide Zertifikate gültig sind, beim Kettenmodell erfolgt keine Korrektur.

[10] Der ausgewiesene Gültigkeitszeitraum ist wie in [9] korrigiert. Wenn ein valides OCSP Prüfergebnis vorliegt und das Zertifikat zurückgezogen wurde, wird das Gültigkeitsende auf den darin ausgewiesen Rückrufzeitpunkt (6.1.6, XML.OCSP, Element <RevocationTime>) korrigiert, sofern er vor dem bereits ermittelten Gültigkeitsende liegt.

<AttribCert Selector>

<Status Info> Int
<OCSPStatus Info Sysdate> Int
<Holder> String
<NotBefore> DateTime
<NotAfter> DateTime
<Serial> String
<Issuer> String
<Certificate> XML.CERTIFICATE
<Procuration> String
<Restriction> String
<AdditionalInformation> String
<Admission> String
<OCSP> XML.OCSP

AttribCert	Container, Beschreibungsblock für ein Attributzertifikat.
Selector	Ein Ident für das Zertifikat, über das die Binärform des Zertifikats über E.APP angefordert werden kann. Beim Export bestimmt dieses den angehängten Anteil des Dateinamen.
Status	Ergebnis der Zertifikatsprüfung nach CERTIFICATE.STATUS
Info	textuelle Darstellung des Status (s.v.)
OCSPStatus	Ergebnis der Online Zertifikatsprüfung nach CERTOCSP.STATUS
Info	textuelle Darstellung des OCSPStatus (s.v.)
Sysdate	Systemdatum zur Einholung der OCSP-Information
Holder	voll qualifizierter Name des Attributinhabers oder des Ausstellers mit Seriennummer eines Zertifikats des Inhabers
NotBefore	Beginn des Gültigkeitszeitraum in UTC
NotAfter	Ende des Gültigkeitszeitraums in UTC
Serial	Seriennummer des Zertifikats
Issuer	voll qualifizierter Name des Ausstellers
Certificate	Container, Ausstellerzertifikat nach XML.CERTIFICATE (sofern verfügbar.
AttributeCertificates	Wert von LiabilityLimitationFlag, default „false“
QualifiiedForSignature	true, wenn das Zertifikat für die Erstellung qualifizierter Signaturen geeignet ist.
Procuration	Angaben zur Vertretungsmacht
Restriction	Selbstbeschränkung, Freitext
AdditionalInformation	Zusatzinformationen, Freitext
Admission	Angaben zur beruflichen Zulassung

OCSP	Container, Daten eines OCSP-Checks nach XML.OCSP, wenn eine OCSP-Check erfolgreich durchgeführt werden konnte

6.1.6 XML.OCSP

Struktur für die Response-Daten eines OCSP-Checks

```
<OCSP Selector>
    <Date> DateTime
    <Status Info> Int
    <RevocationTime> DateTime
    <Reason Info> Int
    <Certificate> XML.CERTIFICATE
```

OCSP	Container, Ergebnisdaten des einer durchgeführten OCSP-Abfrage
Selector	Selektor für die binären Daten der OCSP-Response.
Date	Datum und Uhrzeit aus der OCSP-Response
Status	Zustand des Zertifikats aus der OCSP-Response nach OCSP.STATUS.
Info	textuelle Darstellung des Status (s.v)
RevocationTime	Zeitpunkt des Rückrufs
Reason	Grund des Rückrufs nach REVOCATION.STATUS
Info	textuelle Kurzbeschreibung der Reason (s.v.).
Certificate	Zertifikat des Responders nach XML.CERTIFICATE

6.1.7 XML.TIMESTAMP

Daten zu einem Zeitstempel.

```
<TimeStamp Selector>
    <Status Info> Int
    <Time> DateTime
    <ImprintHashMethod> HASH.METHOD
    <ImprintHashValue> HexString
    <Certificate> XML.CERTIFICATE
```

TimeStamp	Wurzelelement für die Daten zu einem Zeitstempel.
Selector	Ident für den Abruf der Daten des Zeitstempels über LoadObject oder Name der Exportdatei

Status	Ergebnis der Prüfung des Zeitstempels gegen die Signatur nach TIMESTAMP.STATUS
Info	Textuelle Darstellung des Status (s.v.)
Time	Zeitpunkt der Zeitstempelausstellung
ImprintHashMethod	Algorithmus zur Hashwertbildung nach HASH.METHOD
ImprintHashValue	Hex-Darstellung des im Imprint des Zeitstempels gefundenen Hashes
Certificate	Container, Zertifikatsdaten des Responders nach XML.CERTIFICATE

6.1.8 XML.STOREPARAM

Diese Parameterstruktur enthält die erforderlichen Angaben für die Erstellung einer Signatur. Je nach Art und Umfang der Signatur werden unterschiedliche Teile benötigt.

<Params>
- <Description> String
- <XMLSignature ID ParentPath BeforeElement>
  - <Comment> String
  - <Reference URI Type>
    - <EnvelopedSignature>
    - <Base64>
    - <XPath> String
    - <XPathFiltering Filter> String
    - <Canonicalization Comments Exclusive>
- <PKCS7Signature Mime Detached>
  - <sigHint> String
  - <unstructuredName> String
  - <unstructuredAddress> String
  - <emailAddress> String
- <TimeStamp>
  - <URL> String
  - <Userid> String
  - <Pass> String
  - <Policy> String
  - <Digest> HASH.METHOD

Params	Root-Element der Parameter. Hat für sich selbst keine Bedeutung.
Description	Optionale Inhaltsangabe zu dem Dokument. Dieses Element darf nur einmal pro Parametersatz vorkommen. Dieser Text wird in der Detailanzeige zu dem geladenen Dokument angezeigt. Einfache Textformatierungen bleiben nur bei folgender Kodierung erhalten: Zeilenwechsel:
Tabulator:
XMLSignature	Kennzeichnet die Parameter für eine Signatur nach XML-Signature. Muss genau einmal und darf nicht zusammen mit PKCS7Signature auftreten. Dieser Parameter wird nur für die Erstellung einer Signatur herangezogen.
ID	Pflichtattribut zu XMLSignature. Bezeichnet die ID, die der ds:Signature-Knoten tragen soll. Die angegebene ID darf in dem Dokument nicht bereits verwendet sein.
ParentPath	Pflichtattribut zu XMLSignature. Bezeichnet, in welchem Parent-Knoten der Signature-Knoten einzutragen ist. Der angegebene Parent-Knoten muss existieren. Die Prüfung, ob dieser Knoten existiert, wird bei der Signaturausführung durchgeführt. Der Aufbau des Pfads wird unter XML.PATH beschrieben.
BeforeElement	Attribut zu XMLSignature. Gibt das Element im Parent-Knoten an, vor dem der Signature-Knoten eingefügt werden soll. Fehlt die Angabe, wird am Ende eingefügt. Die Zählung beginnt bei 1. Bei Werten < 1 wird der Wert 1 angenommen. Bei Werten > Anzahl der Elemente im Parent-Knoten wird am Ende eingefügt.
Comment	Optionaler Kommentar, der in das ds:SignedInfo-Element eingetragen und mit signiert wird, darf mehrfach vorkommen
Reference	XML-Signature erwartet die Angabe der zu signierenden Daten in Reference-Knoten. Dieser Parameter legt fest, dass ein Reference-Knoten zu bilden ist und legt die Bildungsregeln fest. Die Transformationen werden in der Reihenfolge eingesetzt, wie sie hier angegeben sind.
URI	optionales Attribut zu Reference. Ein String, der in das Attribut URI des Reference-Knotens übernommen wird. Da keine externen Ressourcen vorgesehen sind, muss dieser Parameter mit einem ‚#' beginnen.
Type	optionales Attribut zu Reference. Ein optionaler String, der in das Attribut Type des Reference-Knotens übernommen wird. Dieser Parameter ist beliebig und wird nicht weiter geprüft.
EnvelopedSignature	Fügt die Transformation nach XML-Signature für enveloped signatures ein. Es sind keine weiteren Angaben vorgesehen.
Base64	Eine Transformation eines Base64-Blocks. Für die Signatur eines originären Binär-Datenblocks.
XPath	Fügt eine Transformation nach XPath ein.
XPathFiltering	Fügt eine Transformation nach XPathFiltering ein
Filter	Attribut zu XPathFiltering. Belegung des gleichnamigen Attributs in der zu erstellenden Transformation (Spezifikation

	http://www.w3.org/TR/xmldsig-filter2/). Erlaubt sind „intersect“, „subtract“ und „union“.
Canonicalization	Dieses Element muß am Ende eingefügt werden und bestimmt die Art und Weise der Canonicalization
Comments	Bool. Wenn true, dann werden Kommentare in der Canonicalization berücksichtigt
Exclusive	Bool. Wenn true, wird eine Transformation nach Exclusive-Canonical-XML durchgeführt
PKCS7Signature	Kennzeichnet die Parameter für eine Signatur nach PKCS#7. Muss genau einmal und darf nicht zusammen mit XMLSignature auftreten.
Mime	Bool, optional, Voreinstellung: false. Wenn true, wird ein Ausgabe-format nach S/MIME erstellt
Detached	Bool, optional, Voreinstellung: false. Wenn true, wird die Signatur als detached signature erstellt
sigHint	String, optional. Dieser Parameter ist optional und ist dafür vorgesehen, die applikationsspezifische Rolle zu bestimmen, in der signiert wird. Der Parameter belegt den ersten String in unstructuredName nach PKCS#9 als ASN1-Typ OCTET_STRING. Darf nicht zusammen mit unstructuredName verwendet werden. Dieser Wert wird mitsigniert.
unstructuredName	String, optional. Füllt den ersten String in unstructeredName nach PKCS#9 aus. Darf nicht zusammen mit sigHint gesetzt sein. Dieser Wert wird mitsigniert.
unstructuredAddress	String, optional. Füllt den ersten String in unstructuredAddress nach PKCS#9 aus. Dieser Wert wird mit signiert.
emailAddress	String, optional. Füllt den ersten String in emailAddress nach PKCS#9 aus. Dieser Wert wird mit signiert.
TimeStamp	Kennzeichnet die Parameter zur Integration von Zeitstempeln in die Signaturen. Dieser Knoten ist optional. Wenn er fehlt, wird bei Signaturen kein Zeitstempel angefordert.
URL	String, URL unter der der Zeitstempeldienstanbieter einen Responder nach RFC 3161 über das Protokoll HTTP zur Verfügung stellt.
Userid	String, optional. Benutzername zur Authentisierung gegen den Zeitstempeldienstanbieter
Pass	String, optional. Passwort zur Authentisierung gegen den Zeitstempeldienstanbieter
Policy	String, optional. OID der ggf. vom Zeitstempeldienstanbieter geforderten Policy in Punktnotation
Digest	String, optional. Identifikation des bei der Zeitstempelanfrage zu verwendenden Digests nach HASH.METHOD, vorbelegt mit “SHA256“

6.1.9 XML.PATH

Der Pfad zur Angabe einer Position innerhalb des XML-Dokuments besteht aus wiederholbaren Angaben der Form

Prefix:Elementname[Index] / ...

Elementname	Name des XML-Knotens ohne Präfix. Es wird nach Unterelementen mit diesem Namen gesucht. Alle gefundenen Unterelemente bekommen Indexnummern beginnend bei 1 zugewiesen.
Prefix:	dieser Teil ist optional. Wird er angegeben, wird zusätzlich auf diesen Präfix geprüft. Wird nur der Doppelpunkt ":" angegeben, wird nach Elementen mit leeren Präfix gesucht.
[Index]	dieser Teil ist optional. Wird er angegeben, wird das Unterelement mit der angegebenen Indexnummer verwendet. Fehlt dieser Teil, wird Index mit 1 angenommen. Wird Index < 1 oder größer als die Liste der Unterelemente gewählt, wird die Suche abgebrochen.
/	bedeutet, dass eine weitere Angabe folgt. Die Suche wird in den Unterelementen des vorher gefundenen Elements fortgesetzt. Das Zeichen wird hinter allen Angaben außer der letzten benutzt.

Leerzeichen und Steuerzeichen sind nicht zulässig.

Die Suche beginnt immer vor dem Root-Knoten. D.h. die erste Positionsangabe enthält immer den Namen des Root-Knotens.

6.1.10 DOCUMENT.FORMAT

Datenformat des geladenen Dokuments. Das Format wird zusammen mit dem Dokument und den Metadaten bereitgestellt.

Tabelle 6.2 - DOCUMENT.FORMAT

Typ	**Bemerkung**
P7M	Es wurde ein nach PKCS#7 signiertes Dokument bereitgestellt
P7S	Es wurde eine detached Signature nach PKCS#7 bereitgestellt
XML	Es wurde ein XML-Dokument bereitgestellt
CRT CER DER	X509 Zertifikatsdatei

Alle weiteren Formatangaben werden nur zur Information herangezogen.

6.1.11 DOCUMENT.STATUS

Definition des Verarbeitungsstatus eines Dokuments. status bezeichnet den Status. msg bezeichnet den für de-DE angezeigten Text in E.DISPLAY. Dieser Text wird zusätzlich als Info-Attribut in der XML-Struktur ausgegeben.

Tabelle 6.3 - DOCUMENT.STATUS

Status	Msg	Bemerkung
0	unbearbeitet	das Dokument wurde geladen und es wurden noch keine weiteren Operationen ausgeführt
1	wird signiert	auf diesem Dokument wird gerade ein Signaturvorgang ausgeführt
2	wird verifiziert	das Dokument wird gerade verifiziert
3	signiert	Das Dokument ist signiert, aber nicht verifiziert
4	geprüft	Das Dokument ist verifiziert worden. Es wurde keine Signatur ausgeführt
5	signiert und geprüft	Es wurde eine Signatur ausgeführt und das Dokument ist verifiziert worden.
6	ungültig	Das Urdokument wurde während der Signatur verändert

6.1.12 SIGNATURE.STATUS

Cryptographie-Status einer Signatur. Konvention s.v.

Tabelle 6.4 - SIGNATURE.STATUS

Status	Msg	Bemerkung
0	unbekannt	keine Signatur gefunden / keine Prüfung durchgeführt
1	konsistent	die mathematische Konsistenz der Signatur wurde bestätigt
2	gebrochen	die mathematische Prüfung hat Inkonsistenzen aufgezeigt

6.1.13 SIG.TIMESOURCE

Quelle der ausgewiesenen Signaturzeit

Tabelle 6.5 - SIG.TIMESOURCE

Status	Msg	Bemerkung
0	unbekannt	es liegt keine bekannte Zeitquelle vor
1	Systemzeit	aktuelle Systemzeit
2	Signatur	Angabe aus Signatur, Systemzeit während der Signatur
3	Zeitstempel	Zeitangabe aus gültig geprüftem Zeitstempel

6.1.14 OCSP.STATUS

Status einer OCSP-Response nach RFC 2560. Konvention s.v.

Tabelle 6.6 - OCSP.STATUS

Status	Msg	Bemerkung
0	gültig	Das geprüfte Zertifikat ist im Verzeichnis des Ausstellers vorhanden und nicht zurückgerufen.
1	zurückgerufen	Das geprüfte Zertifikat ist im Verzeichnis des Ausstellers und wurde zurückgerufen
2	unbekannt	Das angefragte Zertifikat ist im Verzeichnis des Ausstellers nicht vorhanden oder nicht veröffentlicht

6.1.15 REVOCATION.STATUS

Beschreibt den Grund des Rückrufs eines Zertifikats nach RFC 2560.

Tabelle 6.7 - REVOCATION.STATUS

Status	Bedeutung (englisch)
0	unspecified
1	keyCompromize
2	cACompromize
3	affiliationChanged
4	superseded
5	cessationOfOperation
6	certificateHold

6.1.16 CERTOCSP.STATUS

Status einer OCSP-Anfrage zu einem Zertifikat. Konvention s.v. Dieser Status berücksichtigt den Umstand, dass bei einer OCSP-Anfrage auch selbst Fehler auftreten können.

Tabelle 6.8 - CERTOCSP.STATUS

Status	Msg	Bemerkung
0	ungeprüft	es ist noch keine Prüfung erfolgt
1	gut	das Zertifikat ist beim Aussteller als gültig bekannt
2	zurückgerufen	das Zertifikat wurde beim Aussteller zurückgerufen (ungültig)

3	unbekannt	das Zertifikat ist beim angefragten Aussteller nicht bekannt
4	abfragefehler	es konnte (temporär) keine gültige Antwort eingeholt werden.
5	aussteller unbekannt	der Aussteller des Zertifikats ist nicht bekannt
6	verifikationsfehler	das Zertifikat des OCSP-Responders konnte nicht verifiziert werden

6.1.17 CERTIFICATE.STATUS

Status zum Prüfergebnis eines Zertifikats. Konvention s.v.

Tabelle 6.9 - CERTIFICATE.STATUS

Status	Msg	Bemerkung
0	ungeprüft	das Zertifikat ist nicht geprüft worden
1	gültig	das Zertifikat ist nach dem Prüfmodell gültig
2	ungültig	das Zertifikat ist nach dem Prüfmodell ungültig
3	vertraut	Zertifikat vertrauenswürdig, im Zertifikatsstamm
4	kettenfehler	die Zertifikatskette ist fehlerhaft oder unvollständig
5	referenzfehler	das Attributzertifikat referenziert nicht das Hauptzertifikat[11]

6.1.18 VERIFICATION.STATUS

Gesamtstatus zum Prüfergebnis von Dokumenten und Signaturen. Konvention s.v.

Tabelle 6.10 - VERIFICATION.STATUS

Status	Msg	Bemerkung
0	Signatur ungeprüft	es steht kein Verifikationsstatus zur Verfügung.
1	Signatur gültig	keine Prüfung hat einen Fehler festgestellt
2	Signatur ungültig	mindestens ein Fehler ist aufgetreten
3	Signaturzertifikat ungültig	das Signaturzertifikat ist ungültig (abgelaufen)
4	Zertifikat nicht mitsigniert	das Signaturzertifikat wurde bei der Signatur nicht mitsigniert
5	Zertifikat nicht gefunden	das Signaturzertifikat ist nicht in der Signatur enthalten oder konnte nicht ermittelt werden

[11] Dieser Zustand wird nur von Attributzertifikaten angenommen und nur genau dann, wenn das Attributzertifikat nicht das Hauptzertifikat (Signaturzertifikat) referenziert. Die Referenz erfolgt über die Struktur Holder des Attributzertifikats.

6	Zertifikat für Signaturen ungeeignet	das Zertifikat ist zur Erstellung von Signatur nicht geeignet
7	Signaturzertifikat unbekannt	das Signaturzertifikat ist beim Aussteller unbekannt
8	Signaturzertifikat zurückgerufen	das Signaturzertifikat wurde beim Aussteller zurückgerufen
9	Zertifikatsaussteller unbekannt	der Aussteller des verwendeten Zertifikats ist nicht bekannt
10	Sicherheitseignung abgelaufen	ein in der Signatur verwendeter Algorithmus gilt zum Prüfzeitpunkt als unsicher
11	Validierung unvollständig	bei der Validierung der Signatur sind Warnungen aufgetreten, die Gültigkeit kann bestritten werden

6.1.19 TIMESTAMP.STATUS

Gesamtstatus zum Prüfergebnis eines Zeitstempels.

Tabelle 6.11 - TIMESTAMP.STATUS

Status	Msg	Bemerkung
0	ungeprüft	es wurde keine Prüfung durchgeführt
1	gültig	keine Prüfung hat einen Fehler festgestellt
2	ungültig	mindestens ein Fehler ist aufgetreten

6.1.20 SAK.STATUS

Arbeitsstatus der SAK. Dieser Status bestimmt, welche Operationen ausgelöst werden können.

Tabelle 6.12 - SAK.STATUS

Status	Bezeichnung	Bemerkung
0	Reset	Die SAK befindet sich in einem Zustand, der dem unmittelbar nach dem Start entspricht
1	Initialized	Der Integritätstest war erfolgreich, es sind keine Objekte geladen
2	Loading	Es ist mindestens ein Objekt geladen, es wurde noch keine Folgeoperation ausgelöst.
3	Loaded	Die Objekte sind geladen und es wurde eine Folgeoperation ausgelöst. Es wurde keine (erfolgreiche) Signatur durchgeführt. Das Auslösen einer Signatur ist zulässig.
4	Signing	Die geladenen Objekte werden gerade signiert. Dieser Zustand wird automatisch beendet. Währenddessen kann keine weitere Operation

		ausgelöst werden.
5	Confirm	Der Benutzer nach erfolgter Signatur oder Verifikation zur Bestätigung aufgefordert.
6	Signed	Alle geladenen Objekte sind erfolgreich signiert worden. Es kann keine weitere Signatur ausgelöst werden.
7	Verifying	Die geladenen Objekte werden gerade verifiziert. Dieser Zustand wird automatisch beendet. Währenddessen kann keine weitere Operation ausgelöst werden.
8	Checking	Auf den Objekten wird gerade eine Zertifikats- oder Signaturprüfung ausgeführt. Dieser Zustand wird automatisch beendet. Währenddessen kann keine weitere Operation ausgelöst werden.
9	Exporting	Über E.DISPLAY ist ein Export ausgelöst worden
10	Online	Eine Onlineanfrage über die Applikation wird ausgeführt. Dieser Zustand wird automatisch beendet. Währenddessen kann keine weitere Operation ausgelöst werden.

6.1.21 READER.STATUS

Zustand des Kartenlesers bzw. des Treibers

Tabelle 6.13 - READER.STATUS

Status	Bezeichnung	Bemerkung
0	Undefiniert	Der Treiber wurde noch nicht verwendet, oder der Zustand des Kartenlesers ist noch nicht bestimmt worden
1	Bereit	Der Treiber ist geladen, das Kartenlesegerät ist verwendbar
2	Fehler	Es sind Fehler aufgetreten

6.1.22 CARD.STATUS

Zustand der Signaturkarte

Tabelle 6.14 - CARD.STATUS

Status	Bezeichnung	Bemerkung
0	Undefiniert	Der Zustand der Signaturkarte ist noch nicht bestimmt worden
1	SingleClose	Signaturkarte ist bekannt und geschlossen
2	SingleOpen	Signaturkarte ist zur Signatur geöffnet
3	MultiClose	Multisignaturkarte ist bekannt und geschlossen
4	MultiOpen	Multisignaturkarte ist zur Signatur geöffnet
5	Fehler	Es sind Fehler aufgetreten

6.1.23 BINARY.EXPORT

Zu allen XML-Elementen mit dem Attribut „Selector“ ist ein Export von Binärdaten vorgesehen.

Die Binärdaten werden über die Funktion LoadObject von E.APP von der Applikation angefordert oder durch Benutzeranforderung auf E.FILE ausgegeben.

Die Applikation kann über LoadObject einen Datenblock nach XML.CONTENT anfordern und kann so die Selector-IDs ermitteln. Beim Export der Daten werden alle Binärdaten als Dateien bereitgestellt und die Namen der Dateien in den Selector-Attributen hinterlegt. Zusätzlich wird die Datei nach XML.CONTENT erstellt, die alle exportierten Dateien in einen Zusammenhang bringt.

Tabelle 6.15 - BINARY.EXPORT

Element	Format	Bemerkung
Signature	XML/Binär	Erstellt den Datenblock, aus dem der Hashwert gebildet wurde, der der Signatur zugeführt wurde. Bei XML ist das der transformierte SignedInfo-Block und bei PKCS#7 die Rohdaten
Reference	XML	In XML-Signature werden die zu signierenden Daten durch Reference-Strukturen angegeben. Zu jedem Reference-Knoten wird ein eigener Hashwert gebildet. Als Binärdaten wrid der aus den Reference-Informationen gebildete transformierte Datenblock exportiert.
Certificate	X509	Es wird das Zertifikat in seiner X509-Struktur exportiert
OCSP	OCSP	Es wird die OCSP-Response bereitgestellt, so wie sie empfangen wurde.
TimeStamp	TS	Es wird der TimeStamp so bereitgestellt, wie er empfangen wurde bzw. wie er im Dokument hinterlegt wurde.
SignedObject	XML/PKCS#7	Das fertig signierte Dokument, abhängig von der Art des Dokuments und der Art der angefertigten Signatur.
StoredObject	XML/Binär	Das Dokument oder Zertifikat, so wie es mit StoreObject übergeben wurde
StoredSubdata	Binär	Das von einer abgesetzten Signatur (detached signature) nach PKCS#7 referenzierte Dokument. Wird ausschließlich zur Prüfung von detached signatures nach PKCS#7 bereitgestellt.

6.1.24 SIGNATURE.METHOD

Kürzel der bei einer Signatur verwendeten Algorithmen und Parameter. Die Schlüssellänge in Bits wird im Attribut KeyLength des Elements verzeichnet.

Tabelle 6.16 - SIGNATURE.METHOD

Kürzel	Bedeutung
RSA SHA1	Verschlüsselung RSA PKSC#1, Hash SHA-1
RSA SHA224	Verschlüsselung RSA PKSC#1, Hash SHA-224

RSA SHA256	Verschlüsselung RSA PKSC#1, Hash SHA-256
RSA SHA384	Verschlüsselung RSA PKSC#1, Hash SHA-384
RSA SHA512	Verschlüsselung RSA PKSC#1, Hash SHA-512
RSA RIPEMD160	Verschlüsselung RSA PKSC#1, Hash RIPEMD-160

Weitere Signaturmethoden sind vorhanden, jedoch nicht Teil der Bestätigung.

6.1.25 HASH.METHOD

Kürzel des bei der Hashwertbildung eingesetzten Hashalgorithmus

Tabelle 6.17 - HASH.METHOD

Kürzel	Bedeutung
SHA1	Hash SHA-1
SHA224	Hash SHA-224
SHA256	Hash SHA-256
SHA384	Hash SHA-384
SHA512	Hash SHA-512
RIPEMD160	Hash RIPEMD-160

Weitere Hashmethoden sind vorhanden, jedoch nicht Teil der Bestätigung.

6.2 Applikationsschnittstelle

Über diese Schnittstelle kommuniziert die Applikation mit der SAK.

Dieser Schnittstelle liegt folgendes Modell zugrunde:

- Die SAK verfügt über mehrere Ablagefächer, in denen jeweils ein Objekt zur Bearbeitung abgelegt werden kann.
- Alle Daten werden vor der Bearbeitung an die SAK übertragen und nach der Bearbeitung von dort angefordert.
- Nach dem Auslösen einer Bearbeitung können keine weiteren Objekte bereitgestellt werden.
- Für eine neue Signatur müssen die Ablagefächer neu belegt werden. Eine Verifikation kann ohne erneutes Laden der Objekte wiederholt werden.
- Eine Bearbeitungsfunktion wirkt immer auf alle Ablagefächer. Jedes Ablagefach wird über XML.STOREPARAM spezifisch mit Parameterdaten versorgt.

6.2.1 Grundlagen

Die Kommunikation zwischen Applikation und der SAK erfolgt über eine bidirektionale Stream-Schnittstelle. Alle Daten werden serialisiert zur anderen Seite übertragen. Zum Aufbau der Kommunikation führt der aufrufende Prozess eine Umlenkung der Standard-Input-Output-Streams auf eigene Handles durch und hat so eine exklusive Stream-Schnittstelle zur SAK, über die beidseitig Daten übertragen werden können.

6.2.2 TSF-Protokoll

Aufbauend auf der Stream-Schnittstelle findet eine Strukturierung der ein- und ausgehenden Daten nach folgenden Prinzipien statt

- Die Kommunikation erfolgt über eine Sequenz von Feldern
- Jedes Feld wird von dem folgenden durch einen Tabulator (ASCII 09) separiert
- Die Sequenz wird durch ein einzelnes Linefeed (ASCII 10) abgeschlossen
- In den Feldern treten keine Steuerzeichen auf. Bei der Kodierung werden diese in eine Zeichensequenz \xx umgesetzt, wobei xx die hexadezimale Repräsentation des Steuerzeichens ist. Ein im Feldinhalt vorkommender Tabulator würde also als \09 im Stream kodiert. Der Backslash selbst wird einfach verdoppelt.
- Auf der lesenden Seite werden alle Steuerzeichen außer Tabulator und Linefeed überlesen.
- Alle anderen Zeichen werden ohne Veränderung übertragen.
- Durch die Art der Kodierung können in allen Feldern Binärdaten übertragen werden. Dieses ermöglicht auch eine Verschachtelung von Feldsequenzen.
- Textuelle Feldinhalte in ASCII-, ISO-8859- oder UTF-8-Kodierung werden abgesehen vom Backslash unverändert übertragen.
- Sequenzen werden immer vollständig übertragen

6.2.3 Basis-Kommunikationsprotokoll

Für die Realisierung der Funktionen werden für das TSF-Protokoll (Namensgebung: Tab Separierte Felder) folgende Festlegungen getroffen:

- Das erste Feld enthält eine Anfragenummer, um das Ergebnis der Anfrage zuordnen zu können. Die Anfragenummer kann leer bleiben, wenn sichergestellt ist, dass das Ergebnis eindeutig zugeordnet werden kann, z.B. weil die Ergebnisbezeichnung schon ausreicht oder nur ein bestimmtes Ergebnis erwartet wird. Eine vorhandene Anfragenummer wird in der zugehörigen Ergebnissequenz verwendet.
- Das zweite Feld einer Sequenz bezeichnet die Funktion oder das Ergebnis
- Alle weiteren Felder enthalten die Parameter oder die Rückgabewerte, soweit diese vorgesehen sind.

Funktionsaufrufe und Ergebnisse werden prinzipiell immer gleich übertragen. Sie werden nur durch das zweite Feld unterschieden.

Bei der Kommunikation von Ergebnissequenzen wird in dem zweiten Feld ein Kürzel über den Returncode der Funktion mit nachgestelltem Doppelpunkt und eine Wiederholung der Funktion übertragen. Der Returncode kann einen der folgenden Werte annehmen:

- OK
  Die Funktion wurde sachgerecht verarbeitet.
- ERR
  Bei der Funktionsausführung ist ein Fehler aufgetreten. Details über die Funktionsausführung oder die Fehlerursache können abfragbar oder in der Sequenz selbst enthalten sein.

6.2.4 Aktivierungsphase

In der Aktivierungsphase wird zwischen ITSigner und der Applikation eine der folgenden Sequenzen ausgetauscht.

1) Nach fehlgeschlagener Selbstprüfung

1.Feld	2.Feld	3.Feld	4. Feld	Richtung
(leer)	ERR	ITSigner V2.0 / Win32	<Fehlermeldung>	➔ APP

Die Kommunikation wird seitens der SAK abgebrochen, die SAK beendet sich. Die Fehlermeldung gibt ggf. Aufschluss über die Ursache des Abbruchs.

2) Nach erfolgreicher Selbstprüfung der SAK

1.Feld	2.Feld	3.Feld	4. Feld	Richtung
(leer)	OK	ITSigner V2.0 / Win32	<Zufallswert>	➔ APP
(leer)	Activate	<AppName>	<AUTH>	➔ SAK
(leer)	OK:Activate			➔ APP

Die Applikation übergibt den Applikationsnamen und einen Authentisierungswert, der aus dem Zufallswert, dem Applikationsnamen und einem der Applikation bekannten geheimen Schlüssel gebildet wird, zur Prüfung gegen die Konfiguration.

Im Anschluss wird der reguläre Betrieb nach dem Protokoll mit den nachfolgend benannten Funktionen fortgeführt.

3) Bei fehlschlagenden Authentisierung der Applikation

1.Feld	2.Feld	3.Feld	4. Feld	Richtung
(leer)	OK	ITSigner V2.0 / Win32	<Zufallswert>	➔ APP
(leer)	Activate	<AppName>	<AUTH>	➔ SAK
(leer)	ERR:Activate	<Fehlermeldung>		➔ APP

Die Kommunikation wird seitens der SAK abgebrochen, die SAK beendet sich. Die Fehlermeldung gibt ggf. Aufschluss über die Ursache des Abbruchs.

In der Konfigurationsdatei sind weder der Applikationsname, noch das Passwort klarschriftlich hinterlegt. Zur Prüfung der Authentizität der Applikation mit dem Applikationsnamen <AppName> und dem Passwort <AppPass> befinden sich in der Konfigurationsdatei stattdessen folgende Wertepaare:

ID1 := Bin2Hex(SHA1(<AppName> + '\x20' + <AppPass>))

ID2 := Bin2Hex(SHA1(<ID1> + '\20' + <AppName>))

Die Applikation muss das Ergebnis AUTH folgender Berechnung mit dem aus der Einschaltmeldung der SAK erhaltenen Zufallswert zusammen mit ID2 vorlegen:

AUTH = Bin2Hex(SHA1(<ID1> + '\x20' + <Zufallswert>))

Die SAK berechnet zu jedem registrierten Wert ID1 nach derselben Regel den Authentisierungswert AUTH und vergleicht ihn mit dem vorgelegten. Nur wen dieser Vergleich Identität liefert und die vorgelegte ID2 ebenfalls mit dem Wert aus der Konfigurationsdatei übereinstimmt, wird die Applikationsanmeldung akzeptiert

6.2.5 Funktionen der SAK

Die Darstellung der Funktionen sieht vor, dass auf der rechten Seite in Klammern sie Aufrufparameter und auf der linken Seite in Klammern die Rückgabewerte im Erfolgsfall dargestellt werden. Wenn nur die erfolgreiche Durchführung der Operation zurückgemeldet wird, stehen auf der linken Seite keine Parameter. Die Reihenfolge der Felder im TSF-Protokoll bestimmt sich durch die Reihenfolge der Aufruf- oder Rückgabewerte in der Funktionsbeschreibung.

Alle textuellen Parameter und Rückgabewerte werden UTF-8-kodiert erwartet oder bereitgestellt.

Unbekannte Funktionen werden entsprechend dem Basis-Protokoll als Fehler zurückgegeben. Unbekannte Antworten (die einen Doppelpunkt enthalten) werden nicht beantwortet.

Die SAK stellt folgende Funktionen zur Nutzung durch die Applikation bereit:

Activate(AppName, Auth)

Führt die Anmeldung der Applikation durch (siehe Aktivierungsphase 6.2.4)

AppName	Name der Applikation (i.A. Applikationstitel)
Auth	Berechneter Authentisierungswert

Die Funktion liefert bei erfolgreicher Anmeldung den Returncode OK:Activate, sonst ERR:Activate und eine Fehlermeldung. Die möglichen Fehlermeldungen sind:

"3:Authentisierung überfällig"	Authentisierung wurde nicht rechtzeitig durchgeführt
"4:Authentisierung fehlgeschlagen"	Authentisierung fehlgeschlagen

(Results) CheckCertificates (Selection)

Führt eine OCSP-Prüfung der ausgewählten Signaturzertifikate durch. Die Signaturen sind vorher durch VerifyObjects ermittelt worden. Durch die Beschreibungsdatei nach 6.1.1, XML.CONTENT erfährt der Aufrufer, welche Signaturen in dem Dokument vorkommen und kann deren Zertifikate gezielt prüfen lassen.

Selection	Eine Liste mit Paaren aus (Storageindex, Signatureindex), beginnend mit der Anzahl der Tupel. Storageindex bezeichnet das Ablagefach beginnend bei 1. Signatureindex bezeichnet die Nummer der Signatur beginnend bei 1 in der Reihenfolge, wie sie in XML.CONTENT aufgeführt werden. Zur OCSP-Prüfung geladener Zertifikate ist Signatureindex mit 0 zu belegen.
Results	Liste mit den Zertifikatsprüfergebnissen, beginnend mit der Anzahl der Einträge (identisch zu Selection), gefolgt von den Prüfergebnissen. Jeder Feld eines Prüfergebnisses besteht aus dem numerischen Wert, einem Leerzeichen (' ', \x20) und dem Text (Msg) nach CERTOCSP.STATUS.

Die Funktion kehrt nicht unmittelbar zurück und erlaubt während der Bearbeitung weitere Funktionsaufrufe. In diesem Zustand unzulässige Funktionsaufrufe werden mit Fehlern abgewiesen (siehe 6.2.6). Die Funktion liefert im Erfolgsfall den Returncode OK:CheckCertificates mit der Ergebnisliste, sonst ERR:CheckCertificates mit einer der folgenden Fehlermeldungen:

"Operation nicht ausführbar"	Die Funktion wurde in einem anderen Zustand als Loading, Loaded oder Signed aufgerufen
"Threadwechsel nicht möglich"	Es ist bereits ein anderer Auftrag in Bearbeitung
"Selektorliste falsch"	Die Selektorliste enthält eine falsche Anzahl von Einträgen
"Objekt zu Index nicht vorhanden"	Eine Adressierung aus der Liste konnte nicht zugeordnet werden

DropObjects ()

Löscht alle Ablagefächer

Die Funktion kehrt unmittelbar zurück. Zwischenzeitlich werden keine weiteren Funktionen akzeptiert. Bei Aufruf in einem unzulässigen Zustand liefert Sie den Returncode ERR:DropObjects und die Fehlermeldung "Operation nicht ausführbar", sonst OK:DropObjects.

Export(Path)

Führt einen Export aller Ablagefächer durch

Path	Basispfad des Verzeichnisses in das der Export erfolgen soll

Diese Funktion kehrt unmittelbar nach Ausführung zurück. Sie ist für Debugzwecke vorgesehen und wird in der Produktionsversion der SAK nicht verwendet. Sie liefert stets den Returncode OK:Export.

(Properties) GetCardProperties(Readername)

Liefert die Properties der eingesteckten Karte zur Unterstützung der Applikation. Diese Funktion ist nicht Bestandteil der Bestätigung.

Readername	Name des Kartenlesers
Properties	Liste von Name/Wert-Paaren, wie sie von den Funktionen GetCardProps() und GetCardSigGCert() von E.CARD bereitgestellt werden.

Die Funktion kehrt unmittelbar zurück. Es werden zwischenzeitlich keine weiteren Funktionen akzeptiert.

Diese Funktion ist für zukünftige Erweiterungen vorgesehen und wird gegenwärtig nicht verwendet. Sie ist nicht Bestandteil der Bestätigung.

Bei Aufruf in einem unzulässigen Zustand liefert Sie den Returncode ERR:GetCardProperties mit der Fehlermeldung "Operation nicht ausführbar", sonst ERR:GetCardProperties (ohne Fehlermeldung).

(Liste) GetReaderNames()

Liefert eine Liste mit den Namen der Kartenleser, wie sie über E.CARD ermittelt wurde. Die Liste wird in der Initialisierungsphase ermittelt und ist danach konstant. Diese Funktion kann während der Funktionen, die weitere Funktionsaufrufe zulassen, ausgeführt werden.

Liste	Eine Liste mit den Namen der Kartenleser

Die Funktion kehrt unmittelbar zurück. Es werden zwischenzeitlich keine weiteren Funktionen akzeptiert. Sie liefert stets den Returncode OK:GetReaderNames und den Rückgabewert (die aktuelle Liste).

(ObjectList) LoadObject (index, selectorlist)

Liest das durch index bezeichnete Ablagefach aus. Es werden das signierte Dokument, die Zusatzdaten zur Signatur und Verarbeitungsprotokoll bereitgestellt.

index	Nummer des Ablagefachs, beginnend bei 1
selectorlist	Eine Stringliste mit Selektoren, die festlegen, welche Daten bereitgestellt werden sollen. Die Liste beginnt mit der Anzahl der Selektoren, gefolgt von den Selektoren selbst. Ein leerer Selektor liefert nur die Inhaltsangabe. Aus der Inhaltsangabe lassen sich über die Selector-Attribute die Idente für alle weiteren Objekte ermitteln. Einige Selector-Idente sind festgelegt (s.u.).
ObjectList	Die Liste der angeforderten Objekte in der Reihenfolge der Angaben in selectorlist. Die Liste beginnt mit der Anzahl der Einträge, gefolgt von jeweils einem Tripel aus (selectorid, exists, binarydata) pro Eintrag.

Tabelle 6.18 - Vordefinierte Selektoren

ID	Bedeutung
<leer>	Inhaltsangabe im Format XML.CONTENT
syslog	Das System-Protokoll
signedobject	das signierte Dokument
storedobject	der über StoreObject() abgelegte Datenblock
storedsubdata	der über StoreObject() abgelegte Subdatenblock

Die Funktion kehrt unmittelbar zurück. Es werden zwischenzeitlich keine weiteren Funktionen akzeptiert. Bei Aufruf in einem unzulässigen Zustand liefert Sie den Returncode ERR:LoadObject mit der Fehlermeldung "Operation nicht ausführbar". Wenn das mit index bezeichnete Ablagefach nicht existiert, liefert sie ERR:LoadObject mit der Fehlermeldung "Objekt zu Index nicht vorhanden" andernfalls liefert sie OK:LoadObject und die Exporte der selektierten Objekte.

(Status) GetStatus()

Liefert eine Aussage zum Zustand der SAK. Diese Funktion kann während der Funktionen, die weitere Funktionsaufrufe zulassen, ausgeführt werden.

Status	Der ermittelte Zustand der SAK nach SAK.STATUS

Die Funktion kann jederzeit ausgeführt werden und kehrt unmittelbar zurück. Es werden zwischenzeitlich keine weiteren Funktionen akzeptiert. Sie liefert stets den Returncode OK:GetStatus und den Zahlenwert des aktuellen Status.

Quit ()

Beendet die Kommunikation mit der SAK.

Diese Funktion kann jederzeit ausgeführt werden und kehrt unmittelbar zurück. Sie liefert stets den Returncode OK:Quit.

Wenn Daten zur Bearbeitung geladen sind, bietet die SAK dem Benutzer die Möglichkeit, die Beendigung wahlfrei durchzuführen, andernfalls terminiert sie unverzüglich.

SetLanguage(lang)

Legt die Dialogsprache der SAK fest. Diese Funktion wird nur dann ausgeführt, wenn keine Daten zur Bearbeitung anliegen.

lang	Spezifikation der Sprache nach rfc 4646. Über die Sprachkonfiguration können verschiedene Sprachen unterstützt werden. Teil der Bestätigung ist jedoch nur die deutsche Sprache (de-DE). Dieses ist die Voreinstellung beim Start. Es werden nur die konfigurierten Sprachkennungen akzeptiert.

Die Funktion kehrt unmittelbar zurück. Es werden zwischenzeitlich keine weiteren Funktionen akzeptiert. Bei Aufruf in einem anderen Zustand als Initialized liefert Sie den Returncode ERR:SetLanguage mit der Fehlermeldung "Operation nicht ausführbar", sonst OK:SetLanguage.

SignObjects(Readername, Objektanzahl)

Führt die Signaturerstellung aus. Die Parameter für die Signaturerstellung wurden vorher bei StoreObject mitgegeben.

Readername	Name des Kartenlesers, wie er vorher über GetReaderNames() bekannt gemacht wurde. Es werden nur die bekannten Kartenleser akzeptiert.
Objektanzahl	Optionale Anzahl der für diesen Signaturvorgang vorgesehenen Objekte. Die angegebene Objektanzahl kann die Anzahl der geladenen Dokumentblöcke übersteigen.

Die Optionale Objektanzahl dient der Vorbelegung möglicher Signaturen bei Einsatz von Multisignaturkarten, wenn die der Applikation bekannte Anzahl auszuführender Signaturen die Anzahl der geladenen Objekte übersteigt.

Die Funktion kehrt nicht unmittelbar zurück und erlaubt während der Bearbeitung weitere Funktionsaufrufe. In diesem Zustand unzulässige Funktionsaufrufe werden mit Fehlern abgewiesen (siehe 6.2.6).

Als Returncode liefert sie nur dann OK:SignObjects, wenn auf allen Objekten eine Signatur ausgeführt werden konnte, sonst ERR:SignObjects mit einer der folgenden Fehlermeldungen:

"Operation nicht ausführbar"	Die Funktion wurde in einem anderen Zustand als Loading oder Loaded aufgerufen
"Threadwechsel nicht möglich"	Es ist bereits ein anderer Auftrag in Bearbeitung
"Kein Objekt vorhanden"	Kein Ablagefach allokiert, kein Okjekt geladen
"Parameterblock fehlerhaft"	Einer der übergebenen Paramerblöcke ist fehlerhaft
"Kartenleser nicht verfügbar"	Der angegebene Kartenleser ist nicht verfügbar
"Signaturkarte nicht verfügbar"	Der angegebene Kartenleser enthält keine Signaturkarte
"Zertifikat nicht verfügbar"	Das Signaturzertifikat konnte nicht gelesen werden
"Sicherheitseignung abgelaufen"	Die Sicherheitseignung der mit der Signaturkarte erzeugbaren Signaturen ist abgelaufen
"Zertifikatsnutzung ohne Signatur"	Dem Zertifikat fehlt die Markierung ist zur Nutzung in Signaturen
"Zertifikat nicht vertrauenswürdig"	Das Zertifikat ist nicht vertrauenswürdig. Es konnte keine Zertifikatskette mit bekannter Wurzelinstanz ermittelt werden.
"Zertifikat noch nicht gültig"	Das Zertifikat ist noch nicht gültig
"Zertifikat nicht mehr gültig"	Das Zertifikat ist nicht mehr gültig
"Signaturausführungsfehler"	Fehler bei der Signaturausführung aufgetreten
"Freigabe von Benutzer verneint"	Der Benutzer hat die Übergabe der Ergebnisdaten an die Applikation abgelehnt
"Unbekannter Ausnahmefehler"	Bei der Signaturausführung ist ein unbekannter Ausnahmefehler aufgetreten

StoreObject(Metadaten, Datenformat, Datenblock, Parameter, Subdatenblock)

Speichert den Datenblock des Dokuments zusammen mit den Metadaten und Parametern im nächsten verfügbaren Ablagefach.

Metadaten	Ein String, der den Inhalt des Dokuments eindeutig erkennen läßt.
Datenformat	Angabe der Applikation, in welchem Datenformat der Datenblock angeliefert wird. Siehe DOCUMENT.FORMAT. Wenn diese Angabe fehlt, versucht die SAK das Format festzustellen. Wird das Format angegeben, wird der Datenblock gegen das angegebene Format geprüft, sofern das Format festgelegt ist.
Datenblock	Die Daten für die Erstellung oder Prüfung der Signatur im Format nach PKCS#7, XML, X509-Zertifikat oder unbestimmt. Das unbestimmte Format ist nur für die Signatur nach PKCS#7 geeignet. Die Erkennung des Datenformats erfolgt innerhalb der SAK.

Parameter	Ein XML-Datenblock, der weitere optionale Parameter enthält, die innerhalb der Folgeoperationen auf den Daten benötigt werden. s. XML.STOREPARAM
Subdatenblock	Bei PKCS#7 muss zur Verifikation einer abgesetzten Signatur (detached signature) das Bezugsobjekt mitgeliefert werden. Der Parameter wird nur in diesem Fall benötigt. In allen anderen Fällen bleibt der Parameter leer.

Die Funktion kehrt unmittelbar zurück. Es werden zwischenzeitlich keine weiteren Funktionsaufrufe akzeptiert. Sie liefert den Returncode OK:StoreObject, wenn der Parameterblock leer oder wohlgeformt ist und das Objekt in einem Ablagefach gespeichert werden konnte. Sonst liefert sie den Returncode ERR:StoreObject mit einer der Fehlermeldungen:

"Operation nicht ausführbar"	Die Funktion wurde in einem anderen Zustand als Initialized oder Loading aufgerufen
"Allokationsfehler"	Das Ablagefach konnte nicht allokiert oder die Daten nicht darin gespeichert werden
"Parameterblock fehlerhaft"	Der übergebene Paramerblock ist fehlerhaft

VerifyObjects()

Führt die Prüfung der Signaturen durch. Bei XML werden alle vorhandenen Signaturen ermittelt und einer mathematischen Prüfung unterzogen. Bei PKCS#7 werden die enthaltenen Datenblöcke nach weiteren PKCS#7-Signaturen untersucht. Die Ergebnisse der Prüfung werden zum Objekt im Ablagefach gespeichert und können mit LoadObject() abgefragt werden.

Die Funktion kehrt nicht unmittelbar zurück und erlaubt während der Bearbeitung weitere Funktionsaufrufe. In diesem Zustand unzulässige Funktionsaufrufe werden mit Fehlern abgewiesen (siehe 6.2.6). Sie liefert den Returncode OK:VerifyObjects wenn jedes der geladenen Objekte über mindestens eine Signatur verfügt und alle lokalisierten Signaturen verifiziert werden konnten. Sonst liefert sie ERR:VerifyObjects mit einer der folgenden Fehlermeldungen:

“Operation nicht ausführbar"	Die Funktion wurde in einem anderen Zustand als Loading, Loaded oder Signed aufgerufen
"Threadwechsel nicht möglich"	Es ist bereits ein anderer Auftrag in Bearbeitung
"Verifikationsfehler"	Für mindestens ein geladenes Objekt konnte keine Signatur gefunden werden oder mindestens eine Signatur ist fehlerhaft

Die Applikation stellt folgende Funktionen zur Nutzung durch die SAK bereit

(Response) GetFile (Message, Wildcard)

Veranlasst einen Dateizugriff über die Applikation.

Message	Bezeichnung des erwarteten Dateiinhalts.
Wildcard	Suchkriterium zum Auffinden der Datei (Posix-ähnlich).
Response	der Dateiinhalt, sofern vorhanden.

Die Funktion wird von der Applikation mit OK:GetFile und dem Ergebnis oder ERR:GetFile und einer Fehlermeldung beantwortet.

(Response) GetUrl (URL, Parameter)

Veranlasst einen Netzwerkzugriff über die Applikation. Das Ergebnis der Anfrage wird einer Plausibilitätsprüfung unterzogen. Die Funktion wird derzeit nicht verwendet. Sie ist für zukünftige Anpassungen oder Erweiterungen vorgesehen.

URL	Die Internet-Adresse der Ressource, die angefordert wird.
Parameter	enthält einen Textblock mit Parametern zu einer http-Anfrage. Dieser Text ist so gestaltet, dass er 1:1 in den Header der http-Anfrage übernommen werden kann.
Response	der Datenblock, wie er aus der http-Anfrage zurückkommt.

Die Applikation muss aus der URL, den übergebenen Parametern und seinen eigenen Parametern eine http-Anfrage zusammenstellen und diese an den in der URL angegeben Server weitergeben.

Die Funktion wird von der Applikation mit OK:GetUrl und dem Ergebnis oder ERR:GetUrl und einer Fehlermeldung beantwortet.

Diese Funktion ist für zukünftige Erweiterungen vorgesehen und wird gegenwärtig nicht verwendet.

(Response) PostUrl (URL, Parameter, Datenblock)

Veranlasst einen Netzwerkzugriff über die Applikation mit dem angegebenen Datenblock. Das Ergebnis der Anfrage wird einer Plausibilitätsprüfung unterzogen. Diese Funktion wird für die OCSP-Prüfung verwendet.

URL	Die Internet-Adresse der Ressource, über die die Anfrage erfolgen soll
Parameter	enthält einen Textblock mit Parametern zu einer http-Anfrage. Dieser Text ist so gestaltet, dass er 1:1 in den Header der http-Anfrage übernommen werden kann.
Datenblock	Ein beliebiger Datenblock, der an das Ziel übermittelt werden soll
Response	der Datenblock, wie er aus der http-Anfrage zurückkommt.

Die Applikation muss aus der URL, den übergebenen Parametern, seinen eigenen Parametern und dem Datenblock eine http-Anfrage zusammenstellen und diese an den in der URL angegeben Server weitergeben.

Die Funktion muss von der Applikation mit OK:PostUrl und dem Ergebnis oder ERR:PostUrl und einer Fehlermeldung beantwortet werden.

6.2.6 Statusübergänge

Im Folgenden werden die Zustände und deren Übergänge beschrieben.

Abbildung 6.1 - Statusübergänge

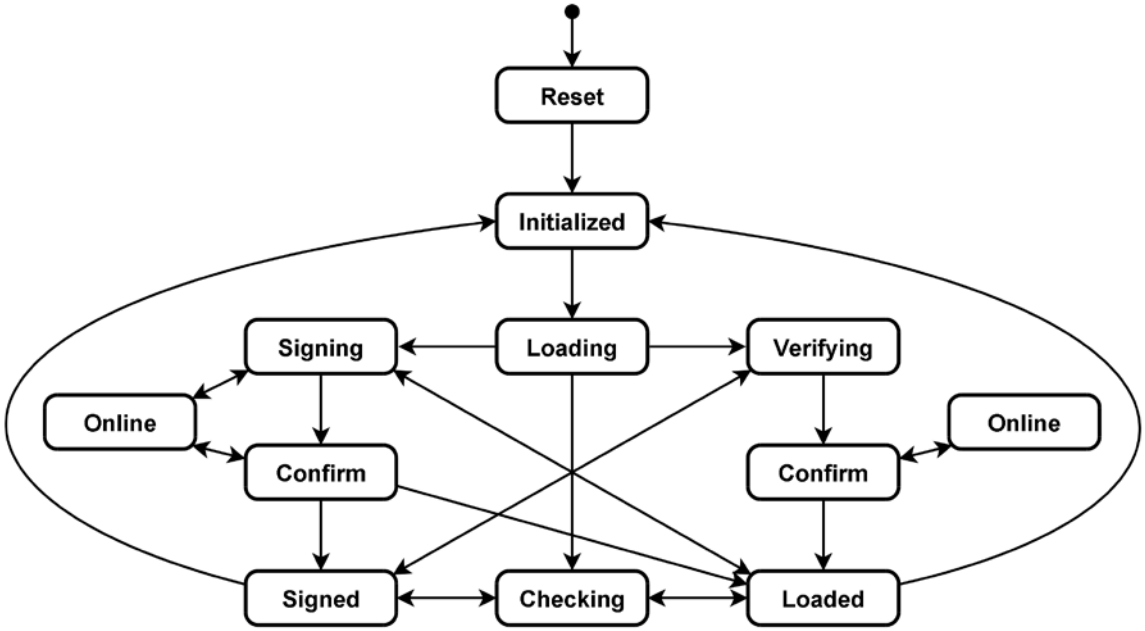


Tabelle 6.19 - Statusübergänge

Status n	Operation	Status n+1	Bemerkung
-	Programmstart	Reset	Anfangszustand
Reset	Initialisierung und Integritätsprüfung	Initialized	automatischer Übergang nach erfolgreicher Integritätsprüfung
*	Quit	-	Es kann jederzeit ein Quit ausgeführt werden
*	GetStatus	*	Der Status kann jederzeit abgefragt werden und verändert den Status nicht
*	GetReaderNames	*	Die Liste der Kartenleser kann jederzeit erfragt werden. Bei nicht durchgeführter Initialisierung ist die Liste leer.
Initialized	DropObjects	Initialized	bewirkt effektiv nichts, ist aber zulässig
	SetLanguage	Initialized	Die Sprache kann nur gewechselt werden, wenn die SAK ordnungsgemäß initialisiert ist und noch kein Objekt zur Bearbeitung geladen wurde.
	StoreObect	Loading	

Loading	StoreObject	Loading	
	LoadObject	Loading	
	SignObjects	Signing	
	VerifyObjects	Verifying	
	Checking	Loaded	
Signing[12]	-	Loaded	automatischer Übergang bei nicht erfolgreicher Signatur. Vorliegende Signaturergebnisse werden gelöscht und die Objekte in ihren ursprünglichen Zustand versetzt.
		$Confirm_S$	automatischer Übergang bei erfolgreicher Signatur
$Confirm_S$[12,13]	-	Loaded	Benutzer hat Übergabe des Signaturergebnisses verneint. Vorliegende Signaturergebnisse werden gelöscht und die Objekte in ihren ursprünglichen Zustand versetzt.
		Signed	Benutzer hat Übergabe des Signaturergebnisses bejaht
Signed	LoadObject	Signed	
	DropObject	Initialized	
	VerifyObjects	Verifying	
	Checking	Signed	
Loaded	SignObjects	Signing	
	DropObjects	Initialized	
	VerifyObjects	Verifying	
	LoadObject	Loaded	
	Checking	Loaded	
Verifying	-	$Confirm_V$	automatischer Übergang nach abgeschlossener Verifikation
$Confirm_V$[12,14]	-	Loaded	Wenn vorher auf den Objekten keine Signatur ausgeführt wurde
		Signed	Wenn vorher auf den Objekte eine Signatur ausgeführt wurde

Die beiden Zustände $Confirm_S$ und $Confirm_V$ unterscheiden sich aus Applikationssicht nur durch die vorangegangene Operation. Beiden gemein ist der Wartezustand auf eine

[12] Aus diesem Zustand kann die SAK kurzfristig in den Zustand „Online“ übergehen, wenn externe Daten benötigt werden (Zeitstempel, OCSP-Anfrage, Laden eines Attributzertifikats). Sie nimmt anschließend wieder den vorangegangenen Zustand an.

[13] Dieser Zustand wird nach erfolgreich ausgeführter Signatur automatisch angenommen. Die SAK zeigt einen Bestätigungsdialog mit der Benutzerwahl, ob das Ergebnis zur Übertragung an die Applikation freigegeben werden soll.

[14] Dieser Zustand wird nach ausgeführter Verifikation automatisch angenommen. Die SAK zeigt einen Bestätigungs-dialog zur Kenntnisnahme des Ergebnisses.

Benutzeraktion (Bestätigung) und die Möglichkeit einer Online-Anforderung aus der SAK. Dieser Zustand dient der Umsetzung der Anforderungen aus dem Signaturgesetz.